

**Universidad de las Ciencias Informáticas  
Facultad 7**



**Universidad de las Ciencias  
Informáticas**

***Título: Herramienta para la gestión de la información  
del Expediente de Proyecto***

***Trabajo de Diploma para optar por el título de  
Ingeniero Informático***

***Autores: Dariel Fernando Reyes Prieto  
Arian Seguí García***

***Tutores: Ing. Yurién R. Fuentes Guerra  
Ing. Yanisbel González Hernández***

***Ciudad de La Habana, Junio 2009***

***“Año del 50 aniversario del triunfo de la Revolución”***

**DECLARACIÓN DE AUTORÍA**

*Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmamos la presente a los 15 días del mes de Junio del año 2009.*

\_\_\_\_\_  
**Autor: Arian Segui Garcia**

\_\_\_\_\_  
**Autor: Dariel Fernando Reyes Prieto**

\_\_\_\_\_  
**Tutor: Ing. Yurién R. Fuentes Guerra**

\_\_\_\_\_  
**Tutor: Ing. Yanisbel González Hernández**

## *Agradecimientos*

*A mi papá y mi mamá por estar conmigo en cada momento de mi vida*  
*A mi hermano por ser más que un hermano*  
*A Mima, mi querida abuela que aunque no este conmigo en este momento siempre estará en mi corazón*  
*A tía Tania por ser madre, hermana y amiga*  
*A tío Juan Andrés, pipito, por ser más que mi tío, mi otro hermano*  
*A mi Abuela Clemencia por darme su amor teniendo tantos nietos*  
*A mis tías Celia y Georgina por ser siempre tan preocupadas por mí*  
*A todos mis tíos y primos de Camagüey por ser mi familia*  
*A todas mis grandes amigas Raque, Niurka y Yaima por estar ahí cuando realmente se necesita*  
*A mi queridísima Jenny, antes novia, hoy y por siempre amiga por aguantarme tanto*  
*A mis amigos de hoy y espero que siempre Leshter, El Pikiri, Duniesky, Romel, Iduany y Alexis.*  
*A todos los que fueron mis profesores por contribuir a mi formación*  
*A mis queridos profes del proyecto, especialmente a Jaque por soportarme siempre*  
*Al profe Cesar y a la profe Idélsis por estar siempre que los necesito*  
*Y a mi amigo y compañero de tesis Seguí por hacer posible este trabajo entre tantas broncas y discusiones.*

*Daniel*

*A mi mamá y mi papá por ser mi luz guía y enseñarme a querer*  
*A Yiska por ser hermana y madre conmigo*  
*A Kikito por ser mi hermano, confidente y mejor amigo*  
*A Yumis por ser una hermana más para mí*  
*A una de las personitas que más quiero con todo el amor de mi corazón, Niurka*  
*A mis grandes amigas Raque, Jenny y Yaima por soportarme todos estos años*  
*A Daryanis y Tania por permitirme su amistad*  
*A Lesther, Pikiri, Iduany, Duniesky y Romel por apoyarme todo este tiempo*  
*A Turin, Ariel y el resto de la pandilla, que nuestros sueños se hagan realidad*  
*Y a mi compañero de Tesis, Daniel, por todas las peleas que tuvimos para lograr este trabajo.*

*Arían*

*A mi familia, especialmente a mi papá, a mi mamá,  
a mi hermano, a mi tía Tania, a mi tío Pipito y a  
mi abuela que está en el cielo y que hizo tanto para  
que yo fuera la persona que soy hoy.  
A todos por darme amor, cariño y comprensión.*

*Daniel*

*A mi mamá, mi papá y mis hermanos,  
por ayudarme a crecer*

*Arián*

## **RESUMEN**

*En la industria de software se necesita mejorar con creces el proceso de gestión de la información que se realiza en los procesos productivos, ya que la calidad no es solo cuestión del producto sino también del proceso de elaboración. La Gestión Documental en la producción de software constituye un paso muy importante para alcanzar una mejor calidad.*

*La Universidad de las Ciencias Informáticas ha desarrollado mecanismos para mejorar dicho proceso, se confeccionó un Expediente de Proyecto (EP) con el objetivo de documentar el producto y su elaboración. Este expediente ofrece significativas ventajas, tanto para el grupo de desarrollo como para los clientes, pero su elaboración resulta engorrosa porque incluye numerosos documentos. Además recoge gran cantidad de información redundante que hace difícil su utilización.*

*Para mejorar la gestión del EP así como su confección, se desarrolló una estructura basada en Esquemas XML; surgiendo la necesidad de una herramienta para el trabajo con los esquemas elaborados. El presente trabajo de diploma tiene como objetivo desarrollar una aplicación que gestione la información del EP a partir de la estructura desarrollada.*

*Con el uso de este sistema, se facilitará el trabajo para los encargados de documentar el proceso de desarrollo software en los proyectos productivos. La aplicación evita la repetición innecesaria en el llenado de datos, validación de la estructura del documento contra los Esquemas XML diseñados y elimina irregularidades en la elaboración del EP. Además garantiza la entrega en tiempo de la documentación y el fácil acceso a la información.*

**Palabras Claves:** Gestión de la Información, Gestión Documental, Expediente de Proyecto, Esquemas XML, Documentos XML.

## Índice

<b>Introducción.....</b>	<b>1</b>
<b>1. Capítulo I. Fundamentación Teórica .....</b>	<b>6</b>
1.1. Gestión Documental .....	6
1.2. Expediente de Proyecto.....	18
1.3. Análisis crítico de las soluciones existentes.....	20
1.4. Tecnologías y herramientas .....	22
1.5. Entorno de desarrollo .....	27
1.6. Metodologías de Desarrollo de Software .....	33
1.7. Selección de las tecnologías a usar .....	38
<b>2. Capítulo II. Características del Sistema .....</b>	<b>40</b>
2.1. Objeto de Automatización.....	40
2.2. Objetivos estratégicos de la UCI como organización .....	40
2.3. Análisis crítico del proceso actual de la situación problemática .....	41
2.4. Descripción de términos comunes.....	42
2.5. Modelo de Dominio.....	42
2.6. Propuesta del Sistema .....	44
2.7. Especificación de Requisitos .....	44
2.8. Definición de los Casos de Usos del Sistema (CUS) .....	47
2.9. Casos de Usos Extendidos.....	51
2.10. Arquitectura del Sistema.....	59
2.11. Características finales del sistema .....	60
<b>3. Capítulo III. Análisis y Diseño.....</b>	<b>62</b>
3.1. Flujos de Trabajo de Análisis y Diseño .....	62
3.2. Modelo del Análisis.....	63
3.3. Modelo del Diseño.....	65
3.4. Patrones.....	66
3.5. Diagramas de clases del Diseño .....	68
3.6. Descripción de las clases .....	69

<b>4. Capítulo IV. Implementación.....</b>	<b>72</b>
4.1. Diagrama de Despliegue .....	72
4.2. Diagramas de componentes.....	72
4.3. Implementación por Módulos.....	73
4.4. Herramienta para la Gestión de la Información del Expediente de Proyecto (GIEP).....	85
<b>Conclusiones .....</b>	<b>86</b>
<b>Recomendaciones.....</b>	<b>87</b>
<b>Referencias Bibliográficas.....</b>	<b>88</b>
<b>Bibliografía.....</b>	<b>91</b>

## **Introducción**

En los últimos años, las tecnologías han conseguido un gran protagonismo en la comunidad científico-tecnológica y, al mismo tiempo, la informática dentro de las ciencias que apoyan ese crecimiento ha aportado cambios significativos. Su impacto a nivel global se ve reflejado en todas las esferas de la sociedad y juega un papel fundamental en la vida del hombre.

Casi la totalidad de los trabajos, trámites, procesos que se realizan en la actualidad están vinculados a la informática, que se convierte en una nueva fuente económica para muchos países. Esto ha propiciado que los avances tecnológicos centren sus objetivos en mejorar la calidad y ampliar sus aplicaciones, principalmente en la producción de software. Los conocimientos actuales están encaminados a lograr que el desarrollo de la informática se realice con la eficiencia requerida para que sea factible su aplicación. La industria del software va en ascenso por lo que la realización de nuevas aplicaciones con múltiples funcionalidades y usos, es cada vez mayor, por lo que resulta una necesidad imperiosa aumentar calidad en la producción.

Para el desarrollo de las aplicaciones informáticas existen numerosas metodologías que se clasifican en ágiles o robustas. El uso de una u otra no influye tanto, sino lo que se realice para la gestión de la información que se cree en este proceso. La mayoría de las veces, constituye más un problema el propio proceso de gestión de la información que la producción del software. El exceso de información constituye uno de los principales problemas y se necesita invertir más tiempo en la misma. Es frecuente encontrar un gran número de información redundante e innecesaria mezclada con la que es realmente importante.

Para las grandes empresas de producción de software que trabajan con gran cantidad de información, les resulta difícil lograr una adecuada gestión de la misma, y por consiguiente alcanzar su calidad, afectando en sentido general su producción. Con el objetivo de garantizar el manejo adecuado de la información surge el término Gestión Documental (GD): consiste en el uso de tecnologías y procedimientos que permitan la gestión y el acceso unificado a la información generada en la organización. La GD proporciona una solución integral para la manipulación y protección de aquellos documentos que se desean preservar en los proyectos como soporte de su negocio. La aplicación de estos procesos no solo hará posible disciplinar a

los involucrados, permitirá además, almacenar y disponer de los datos históricos necesarios para lograr un trabajo más predecible y eficiente. (1)

Este método es aplicable en muchas de las ramas de la ciencia y la técnica, ya que siempre existe documentación generada producto de algún proceso o acción que se realice, llegando a convertirse en una prueba documental del trabajo. En la Industria del Software se genera gran cantidad de información por lo que el uso de métodos de gestión documental se hace indispensable. La eficiencia y la calidad de la producción de software dependen en casi su totalidad de este proceso, convirtiéndose en una meta común lograr estos objetivos.

En muchos países la industria de software se ha convertido en uno de los renglones de la economía que más aporta a su desarrollo, e incluso ha llegado a ser un elemento clave en la situación política y social. Cuba es uno de los países que posee grandes perspectivas en el mundo de la producción de software. La informática en el país se está convirtiendo en una fuente de desarrollo económico-social y para esto se viene realizando un trabajo significativo en materia de la informatización de la sociedad. La formación y capacitación de nuevos profesionales en las diferentes ramas de la informática constituyen un eslabón fundamental en el desarrollo de la industria del software a nivel nacional. El crecimiento vertiginoso de la producción internacional de software requiere procesos de mayor calidad y productividad.

La Universidad de las Ciencias Informáticas (UCI), surgida en el año 2002 en el marco de la Batalla de Ideas con el objetivo de ayudar e impulsar el desarrollo de software en el país, se ha convertido en un centro por excelencia de referencia en la industria de software cubano. Con su poco tiempo de creación ya se han desarrollado una serie de normativas rectoras de los software producidos, todos con buenos resultados, por lo que constituyen un reto en cuanto a organización y calidad en el proceso de producción.

La calidad debe estar de forma permanente en los productos y procesos que se realicen. La introducción con urgencia de estándares y normas en las empresas nacionales constituye actualmente una prioridad. Calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, portabilidad, usabilidad, seguridad e integridad. La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía

de trabajo, en aras de lograr una mayor confiabilidad y facilidad de prueba. (2) La calidad no es solo una cuestión de producto, depende también de la información que lo antecede y representa en todo momento.

Para poder lograr mejores productos en el proceso de desarrollo de software se hace necesario que la gestión de documentos generados se realice con mayor eficiencia y calidad. Es indispensable registrar y documentar todo lo que sucede en cada fase del proceso, de ello dependerá el éxito de cada proyecto. De esta forma se facilitará al cliente y al propio equipo de desarrollo conocer la situación en cada momento del avance obtenido. Además se podrán gestionar transformaciones en el producto de una manera eficiente y rápida sin la necesidad de afectar otras fases del desarrollo de software, aunque exista una relación entre cada una de ellas.

Al inicio de la creación de la UCI como centro productor de software, en sus proyectos productivos la documentación generada se elaboraba de forma independiente, solo registrando aspectos importantes dentro del desarrollo de un producto. Esto no garantizaba la total calidad del proceso de desarrollo de software, entorpeciendo el propio mantenimiento de los productos como también la implementación de futuras versiones.

Como parte de las acciones que permiten mejorar la calidad de los procesos de desarrollo de software, la estandarización de la documentación de los proyectos y el adiestramiento de los equipos de proyecto en el uso de modelos y estándares propios para elevar la calidad del producto entregado al cliente, se diseñó un **Expediente de Proyecto (EP)**, que ya ha comenzado a implantarse en cada uno de los proyectos vigentes en la universidad. Este esquema de expediente y grupo de *plantillas tipo* definidas tiene como objetivo influir en la estandarización de la documentación y la creación de una cultura de calidad en la organización. 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.

La confección del EP resulta trabajosa por la cantidad de información que se registra y el tiempo que requiere su elaboración, llegándose a recoger información redundante que dificulta su elaboración y provocando la inconformidad de los desarrolladores, afectándose la calidad de la documentación del proyecto y retrasos en la entrega de la misma. A partir de estos problemas y de algunas experiencias de los proyectos con el uso del EP, se determina analizar la manera en que se podría agilizar y mejorar este proceso. Para ello se elabora un método estructurado en **Esquemas<sup>1</sup> XML<sup>2</sup>**, surgiendo la necesidad de una herramienta capaz de gestionar la información recogida en la nueva estructura y que de solución a las problemáticas del proceso de GD del EP.

Teniendo en cuenta la situación anterior, el **problema** radica en: La no existencia de una herramienta para gestionar la Información del Expediente de Proyecto basada en Esquemas XML. El **Objeto de estudio** se centra en el proceso de gestión de la documentación de los proyectos productivos en la Universidad de las Ciencias Informáticas. Se determina como **Campo de acción** de la investigación, el Expediente de Proyecto en la Universidad de las Ciencias Informáticas.

Para dar solución al problema antes mencionado se define como **Objetivo general** Desarrollar una herramienta para gestionar la Información del Expediente de Proyecto basada en Esquemas XML. Para llevar a cabo el objetivo del trabajo de diploma se proponen las **Tareas de la Investigación** siguientes:

- Analizar los diferentes sistemas que existen para la gestión de la Información del Expediente de Proyecto y para la gestión de información mediante protocolos basados en XML.
- Evaluar las tecnologías y librerías existentes que puedan ser usadas en el trabajo.
- Seleccionar la Metodología, lenguaje de programación, plataforma y Entorno Integrado de Desarrollo (IDE) para elaborar la herramienta.
- Realizar un análisis sobre la estructura del Expediente de Proyecto en la UCI.
- Describir las problemáticas que existen para la gestión de la información del Expediente de Proyecto en la Universidad.

---

<sup>1</sup> (**XML Schema**) Es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML.

<sup>2</sup> Siglas en inglés de *Extensible Markup Language* («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C)

- Hacer un análisis crítico y valorativo de los Esquemas XML propuestos para ser utilizado por la herramienta.
- Elaborar la documentación correspondiente a los Flujos de Trabajo propuestos por la Metodología seleccionada.
- Implementar las funcionalidades requeridas en la aplicación.

El presente trabajo se encuentra estructurado por 4 capítulos que abordan los siguientes contenidos:

En el Capítulo 1 se describe la Fundamentación Teórica. Se plantean los principales conceptos y términos abordados en la investigación, se analiza el estado del arte del tema tratado a nivel internacional, nacional y en la Universidad, de las tendencias, técnicas, tecnologías, metodologías y software usados en la actualidad o en las que se apoya para darle solución al problema planteado.

En el Capítulo 2 se definen las características del sistema y el Objeto de estudio. Se hace un análisis del dominio de la aplicación, se describen los procesos a automatizar para darle solución al problema y se generan documentos referentes a esta fase. También se definen requisitos no funcionales y un prototipo de interfaz externa.

En el Capítulo 3 se realiza el Análisis y diseño del sistema, se define el Modelo de Análisis, Modelo de Clases de Análisis, representaciones gráficas. En el diseño se realizan los Diagramas de Interacción y Diagrama de Clases.

En el Capítulo 4 se realiza la implementación, que contiene el Diagrama de Componentes así como su representación gráfica. Se reflejará una serie de explicaciones de la estructura interna de la aplicación y también códigos de los métodos más relevantes.

## 1. Capítulo I. Fundamentación Teórica

En el capítulo se aborda la importancia de la GD dentro del desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC). Además se analizan algunos software destinados al proceso de GD y se expone el EP como solución en la UCI al proceso de gestión de la información. Se realiza un estudio de la situación en el mundo, el país y en la universidad sobre el desarrollo de herramientas para realizar el proceso de GD y también el por qué de la necesidad de una herramienta para la automatización de EP, así como la justificación de las metodologías y tecnologías a usar para la elaboración de la misma.

### 1.1. Gestión Documental

#### 1.1.1. El desarrollo de la información y conocimiento, papel de la Gestión Documental

Esta es la era de la información, actualmente la información significa poder, tanto económico como político. La sociedad moderna está basada en los principios del desarrollo tecnológico e industrial, el conocimiento es la fuente de poder para todo este desenfrenado proceso que cada día alcanza un nivel más elevado.

Tener la tecnología y los recursos no significa desarrollo, también es importante tener el personal calificado para operar dichas tecnologías o la información necesaria para preparar dichos profesionales. El conocimiento es uno de los recursos estratégicos de la Sociedad de la Información y la base fundamental de la futura Sociedad del Conocimiento que se desea alcanzar. Para poder llegar a ella se debe ser capaz de adquirir y utilizar el capital intelectual del que se dispone, pero también es necesario, y de forma primordial, transferirlo en beneficio de la propia sociedad, para que además de poder ser utilizado por ella, sea un medio que ayude al desarrollo, generando a su vez más conocimiento.

Uno de los procesos que ayudan a perpetuar la información y el conocimiento es la GD. Se entiende por ello al conjunto de normas, técnicas y prácticas usadas para administrar el flujo de documentos de todo tipo en una organización, permitir la recuperación de información desde ellos, determinar el tiempo que los documentos deben guardarse, eliminar los que ya no sirven y asegurar la conservación indefinida de los documentos más valiosos, aplicando principios de racionalización y economía. (3)

Es una actividad casi tan antigua como la escritura, que nació debido a la necesidad de "documentar" o fijar actos administrativos y transacciones legales y comerciales por escrito para dar fe de los hechos. Este tipo

de documentos se plasmaron sucesivamente en tablillas de arcilla, hojas de papiro, pergaminos y papel, cuya gestión se fue haciendo cada vez más compleja a medida que crecía el tamaño de los fondos documentales.

El uso del computador en la GD se inicia en la práctica a partir de las grandes bibliotecas nacionales anglófonas, la Biblioteca del Congreso de los Estados Unidos de América y la British Library, que en los años 60 del siglo XX crean programas de bases de datos conocidos como MARC (Machine Readable Cataloguing) o Catalogación leíble por computador. Poco después se comienza también a usar registros computarizados para inventariar documentación administrativa en soporte papel.

Cuando el uso de las tecnologías de información y comunicación se hizo común en la administración pública y privada, con el inicio de las bases de datos y la aparición de los procesadores de textos y otras aplicaciones ofimáticas, y sobre todo con la llegada del correo electrónico, surgió la necesidad de capturar y conservar también documentos que nacen, viven y mueren en formato electrónico. Conseguir esto representó un nuevo salto en la complejidad y exigencias a los sistemas informatizados y en la forma de pensar de los administradores y archiveros. (4)

Actualmente existen algunos criterios generales a tener en cuenta, un sistema de gestión documental por lo general se refiere a las siguientes áreas: Almacenamiento, recuperación, clasificación, seguridad, custodia, distribución, creación, autenticación.

**Almacenamiento** ¿Dónde se guardarán los documentos?

<b>Recuperación</b>	¿Cómo puede la gente encontrar documentos necesarios? ¿Cuánto tiempo se puede pasar buscándolos? ¿Qué opciones tecnológicas están disponibles para la recuperación?
<b>Clasificación</b>	¿Cómo se organizarán los documentos? ¿Cómo se asegurará que los documentos estén archivados siguiendo el sistema más apropiado?
<b>Seguridad</b>	¿Cómo se evitará la pérdida de documentos, la violación de la información o la destrucción no deseada de documentos? ¿Cómo se mantendrá la información crítica oculta a quién no debiera tener acceso a ella?
<b>Custodia</b>	¿Cómo se decidirá qué documentos conservar? ¿Por cuánto tiempo deben ser guardados? ¿Cómo se procederá a su eliminación (expurgo de documentos)?

<b>Distribución</b>	¿Cómo se distribuirán los documentos a la gente que la necesita? ¿Cuánto se podrá tardar para distribuir los documentos?
<b>Workflow</b>	¿Si los documentos necesitan pasar de una persona a otra, cuáles son las reglas para el flujo de estos documentos?
<b>Creación</b>	¿Si más de una persona está implicada en creación o modificación de un documento, cómo se podrá colaborar en esas tareas?
<b>Autenticación</b>	¿Cómo se proporcionarán los requisitos necesarios para la validación legal al gobierno y a la industria privada acerca de la originalidad de los documentos y cómo se cumplirán sus estándares para la autenticación?

**Tabla 1** Criterios para la GD

Existen múltiples software que realizan el proceso de gestión basados en estos criterios y con características específicas de cada uno de los procesos a los que les dan solución.

- **Adapting Document:** es una herramienta para compartir, distribuir y gestionar la documentación de una organización en formato digital. Es un sistema de GD de tecnología web configurable dentro de una Intranet/Extranet. Posee un archivo electrónico de documentos seguro y centralizado. El sistema es administrable desde navegador, y dispone de funciones para la búsqueda y distribución controlada de documentos. (5)
- **ADOC LTDA:** Soluciones para Gestión Documental - Administración Documental / Digitalización de Archivos y Documentos, digitalización masiva de documentos, captura de índices mediante OCR<sup>3</sup> y código de barras, digitalización de archivos históricos, conversión a distintos formatos, procesos documentales de alistamiento, clasificación, depuración, indexación de las imágenes para reducir los tiempos de búsqueda, asesoría y acompañamiento en implementación de procesos de digitalización. (6)
- **Plataforma de Acceso a Información de ZyIMAGE (IAP):** ofrece a las organizaciones la mejor gama de herramientas para archivar, buscar, gestionar y compartir información electrónica y en papel, así como mensajes de correo electrónico y multimedia.

<sup>3</sup> **Reconocimiento Óptico de Caracteres**, abreviado habitualmente como **OCR** (*Optical character recognition*)

- **Alfresco:** es un sistema de administración de contenidos de código libre / abierto, basado en estándares abiertos y de escala empresarial para Windows y sistemas operativos similares a Unix. Está diseñado para usuarios que requieren un alto grado de modularidad y rendimiento escalable. Alfresco incluye un repositorio de contenidos, un framework de portal web para administrar y usar contenido estándar en portales, una interfaz CIFS<sup>4</sup> que provee compatibilidad de sistemas de archivos en Windows y sistemas operativos similares a Unix, un sistema de administración de contenido web capacidad de virtualizar aplicaciones web y sitios estáticos vía Apache Tomcat y búsquedas vía el motor Lucene. Alfresco está desarrollado en Java.

El módulo de **Gestión de Documentos de ZyIMAGE** es una herramienta directa que le permite gestionar eficazmente todos sus archivos electrónicos (como los creados con las herramientas de Microsoft Office). Puede añadir una amplia gama de funcionalidades de gestión de documentos a sus archivos ZyIMAGE (índices), incluyendo actividades de protección/desprotección y control de versión.

La siguiente lista describe sólo una de las pocas características estándar del **módulo de Gestión de Documentos ZyIMAGE**, que le permiten controlar y optimizar cualquier actividad de gestión de documentos:

- Control manual y automático de la versión.
- Capacidades seguras y gestionables de protección/desprotección.
- Capacidades de integración completa con las bases de datos más comunes, tales como Microsoft Access, Microsoft SQL<sup>5</sup> Server, y Oracle.
- Herramienta ZyFIND para acceder a su archivo de gestión de documentos.
- ZyIMAGE Enterprise Webserver para ofrecer acceso en línea fácil-de-usar para su sistema de gestión de documentos. (7)

---

<sup>4</sup> Sistema de archivos comunes de Internet o sus siglas en Inglés: Common Internet File System.

<sup>5</sup> Lenguaje de consulta estructurado o sus siglas en Inglés: **Structured Query Language**.

- **MICROCOLSA:** Líder en la GD y el Soporte Posventa Administración Documental / Gestión Electrónica. (8)
- **Siam IRS:** Es una gama de software diseñados para la Creación, Distribución, Almacenamiento, Edición, Mantenimiento y Gestión de todo tipo de informaciones, documentos e imágenes a través de tecnología Internet. **Los Software Siam IRS** (Soluciones Inteligentes para Aplicaciones Multimedia) son aplicaciones de **Gestión de Información y Documentos** que el usuario gestiona a través de Internet, por lo que no necesita para su **conocimientos especiales**. (9)
- **102 NovaDoc:** es una empresa comprometida en el área de **Nuevas Tecnologías aplicadas a Soluciones Documentales**, principalmente Software de Gestión de Bibliotecas, aplicaciones web para archivos (ya sean históricos, administrativos, provinciales, médicos, arquitectos,...), Contenidos Digitales y Gestión Electrónica, todos ellos orientados a Redes e Internet vía web.
- **Suite Documental 102,** logra la mejor integración y ofrece soluciones homogéneas, usando los entornos y estándares actuales con mayor proyección como son los desarrollos basados en **tecnología XML, PHP<sup>6</sup> o .NET**, entre otros, que junto a los servidores de bases de datos **Oracle, MS SQL Server y MySQL** logran una variada oferta de servicios funcionales para diversos sistemas de forma óptima y segura, al mismo tiempo que se consigue la estabilidad, potencia y escalabilidad de las aplicaciones. (10)
- **I-DOC:** Que puede integrarse como módulo complementario de cualquier sistema de información ya existente, permitiendo una implantación más rápida y menos conflictiva, así como una evolución más segura. Este software está compuesto por dos módulos que se integran entre sí de forma secuencial. De esta forma el módulo I-SCAN sería el modelo básico, a partir del cual se estructura el siguiente módulo, conocido como I-GES. Grupo Entorno Documental (GES) propone la evolución de la gestión de la información mediante la integración de la documentación física y electrónica en un entorno documental único, inteligente y flexible. Los documentos se tratan de forma segura y confidencial a lo largo de todo su proceso de vida. (11)

---

<sup>6</sup> *Personal Home Page Hypertext Pre-processor*

- **Documentum Enterprise Document Management System 98:** Potente sistema que automatiza y controla el ciclo vital de grandes cantidades de documentos dentro de las organizaciones a través de diferentes plataformas. Documentum 4i gestiona y procesa el contenido de portales corporativos en ambientes web. (12)
- **FileNET Panagon IDM Document Services:** Aporta una serie de programas y herramientas para la gestión de la documentación en las organizaciones de manera distribuida a través de procesos de *workflow* tanto en arquitecturas de cliente/servidor como en ambientes web. (13)
- **DOCS Open EDMS:** Basado en arquitectura cliente/servidor captura, almacena y gestiona la información de múltiples formatos de documentos. Su extensión DOCS Binder provee una herramienta para la creación, gestión y distribución de conjuntos de documentos en formato XML. (14)
- **Lotus Domino.DOC:** Es una extensión de Notes/Domino para la gestión distribuida de documentos aportando potentes funciones de mensajería y trabajo en grupo. Asegura la integridad de la documentación desde su creación, hasta su archivado final en diferentes dispositivos, todo ello aprovechando la potencialidad del trabajo en grupo de Notes/Domino. Existen otras soluciones comerciales con diferentes características en cuanto a sus posibilidades y estructuras, como son Docuware, Novasoft ó Cindoc de Chemdata (15)
- **Hyperwave Information Server:** Hyperwave Information Server complementado con Hyperwave Information Portal para dotar a la empresa de una completa solución de gestión del conocimiento. Hyperwave Information Server (HIS) constituye el motor y la base que permite el manejo de la información.

Entre las características fundamentales que aporta HIS están la autenticación de usuarios, las funciones de publicación y creación de documentos, búsquedas, grupos de discusión, etc. Si sus permisos se lo permiten, los usuarios pueden contribuir a la información y el conocimiento común de la organización publicando sus propios documentos o retocando los existentes. Los documentos pueden ser simples archivos HTML, DOC<sup>7</sup> o en cualquier otro formato o bien pueden ser tipos complejos como clústeres o secuencias. El precio del producto es bastante alto aunque depende del

---

<sup>7</sup> Extensión de los documentos Office Word (.doc)

número de usuarios que lo vayan a utilizar y en cualquier caso es asequible si se tiene en cuenta las empresas a las que va dirigido. (16)

Todas estas empresas y software están relacionadas con la GD y brindan ingeniosas soluciones a las problemáticas con las que se relacionan, siempre basados en el mismo objetivo, garantizar la organización, la seguridad y la calidad de la información que gestiona.

La industria del software es un sector donde el flujo de información es muy grande siendo necesaria la implementación de un proceso para la gestión de la misma; este debe tener buena calidad y eficiencia en cuanto a la rapidez con que se efectúa.

## 1.1.2. Uso en Cuba del Gestión Documental

Este proceso se utiliza en el país de forma masiva en muchos de las ramas de la producción, realizándose de forma específica en cada una de ellas. En los proyectos de producción de software también se realiza, comenzando desde que se inicia la creación del grupo de desarrollo hasta el desarrollo y conclusión de un producto. El proceso de GD es un medio para garantizar la organización y calidad de todos los procesos que se desarrollen, utilizando el EP como medio para la gestión de la información. Este garantiza de una forma organizada la transformación y constante actualización de la información que se recoge en el mismo.

En el país existen algunas herramientas elaboradas para el control del proceso de GD, unas vinculadas a otros procesos. Ejemplo:

**Avila Doc:** Aplicación para el control de documentos. Permite controlar la entrada y salida de documentos y gestionar el tránsito interno de los mismos, constituyendo un archivo digital que agiliza las tareas de registro, búsqueda, reproducción y distribución de los documentos. Este sistema permite ser enlazado a Microsoft Outlook para generar automáticamente tareas a cumplir, notificaciones e información asociada al documento. Ofrece un historial del documento desde que fue creado hasta su eliminación y tiene un **Precio:** **MN:** \$ 2,707.64.

**PAPIRO:** Un sistema de conservación, digitalización, gestión y socialización de información documental para los archivos en Cuba. Es un producto informático de uso libre que emplea herramientas igualmente libres y permite conservar documentación de valor histórico al evitar su manipulación; pues, al digitalizarse

el documento, este se consulta en formato electrónico. Esta versión incluye también la gestión de publicaciones periódicas, especialmente revistas.

La consulta de los documentos, puede ser realizada a través de una eficiente gestión que permite recuperar información por todas las variables de almacenamiento (campos de obligatoria presencia según la norma ISAD-G, Norma Internacional para la descripción archivística, para intercambio internacional), en la base de datos; mientras, la posibilidad de colocar en la red de redes dichas bases de datos y las imágenes de los documentos, no sólo democratiza la accesibilidad documental; sino, que ahorra recursos materiales y tiempo, haciendo más eficaz y eficiente el proceso de búsqueda e investigación documental. Está concebido para correr bajo la plataforma Windows en las versiones 2000 y XP. Los formatos de imagen de los documentos están en PDF<sup>8</sup>, estructura de uso común y gran poder de compresión, y para su consulta en la red se diseñó un método de gestión dinámica con páginas PHP.

El sistema se emplea en el archivo histórico de la ciudad, y si bien fue diseñado para su empleo en repositorios históricos, también puede aplicarse en archivos de gestión, resultando al mismo tiempo un intento pionero por lo menos en Cuba, al vincular la gestión de bases de datos con las imágenes de documentos originales digitalizados, contribuyendo de este modo a su conservación y socialización.

**Babel:** Es un sistema automatizado para el control y gestión de las solicitudes de servicios propios de la Unidad de Traducción del Centro de Información. Este sistema permite organizar, clasificar y almacenar todos los documentos traducidos, que posibilita por una parte la gestión documental de los mismos, con el propósito de facilitar el acceso a la información a distintos usuarios (clientes o traductores) y, por la otra, la utilización eficiente y segura de estos documentos. Es una herramienta de control de los procesos técnicos que desarrolla la unidad porque recoge un grupo de informaciones relevantes sobre clientes y traductores que pueden ir desde un breve perfil de usuario actualizado hasta el conjunto de competencias que posee el traductor que realizará el trabajo.

Existen otras aplicaciones no comercializadas desarrolladas por productores individuales y que son utilizadas para automatizar algunos procesos no solo de desarrollo de software, significando de gran importancia en este último.

---

<sup>8</sup> Formato de documento portátil o sus siglas en Inglés: *Portable Document Format*.

## 1.1.3. Gestión documental de proyectos de producción de software

En la informática el tratamiento de la información y el conocimiento es un tema complicado en la actualidad. Son muchos los sistemas que se desarrollan hoy en día en la Industria del Software, volviéndose la competitividad un factor fundamental en esta industria a nivel internacional. Por ello, es una garantía presentarse al mundo de la producción de software con efectivos métodos y formas de trabajo que demuestren eficacia y alta productividad.

La producción de software está en constante cambio producto de la misma revolución tecnológica por la que atraviesa el mundo en estos días. Son muchos los sistemas que salen al mercado, todos compitiendo por lograr una mayor calidad y eficiencia en el cumplimiento de las funciones para las que fueron creados, siempre respondiendo a las necesidades de clientes cada vez más exigentes.

Actualmente es muy común el término de Ingeniería de Software, disciplina encargada de planificar los procesos de desarrollo y mantenimiento de un software. La Ingeniería de Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadoras, y la documentación asociada requerida para desarrollar, operar y mantener dichos programas. Se conoce también como desarrollo de software o producción de software. (17)

Para realizar de forma exitosa un proyecto de software es necesaria la correcta ejecución de cuatro tipos de funciones:

- Gestión del Proyecto: incluye fundamentalmente la Estimación, Planificación y Seguimiento del proyecto, Organización, Dirección y Gestión de Recursos Humanos.
- Desarrollo Técnico: actividades de Ingeniería del Software a lo largo de todo el ciclo de vida del producto: Análisis, Diseño, Codificación.
- Sistema de Calidad: incluye las actividades de:
  - Validación: construir el producto correcto.
  - Verificación: comprobar si se está construyendo el producto correctamente.
  - Pruebas: verificar si funciona el código.

- Actividades de Garantía de Calidad: asegurar que el producto que se construya posea determinados niveles de calidad.
- Sistema de Gestión de Configuración: incluye principalmente la identificación, organización y control de las modificaciones que sufre el software. (18)

Quedando demostrado que la información que se maneja en un proyecto es muy grande y se necesita de un mecanismo o herramienta para su documentación, esto permite su posterior consulta y análisis por parte del equipo de desarrollo o por el propio cliente. La necesidad de implantación de herramientas para automatizar el proceso de GD es fundamental.

La gestión de forma electrónica facilita la velocidad de trabajo, así como un mejor tratamiento de la información con la que se trabaja, facilitando de esta manera mejorar la calidad del producto obtenido y estar al tanto de cualquier cambio que se produzca tanto por parte del cliente como en el propio software. A continuación se mencionan algunas de las aplicaciones utilizadas indistintamente como herramientas para la GD de proyectos, estas integran muchos de los avances tecnológicos más usados en la actualidad.

## Herramientas para la Gestión Documental de Proyectos

**Sistema de Gestión de Contenidos (CMS):** La diversidad de CMS de código abierto existente es extraordinariamente amplia, cada uno de ellos poseen las funcionalidades generales de cualquier manejador de contenidos, además de características particulares que los identifican. Habitualmente todo el software de código abierto es de acceso libre, sin ningún coste en licencias, lo cual es muy ventajoso en este caso. Muchos de estos CMS son utilizados como herramientas de gestión documental, en dependencia de las necesidades de la organización que lo vaya a utilizar. Permiten la recuperación y reutilización de documentos, y en general de cualquier objeto publicado o almacenado. Presentan control de acceso gestionando permisos aplicados a grupos o individuos. Dentro de los CMS más conocidos se encuentran:

- **Plone:** Permite que se creen y adicionen diferentes tipos de contenido; todos los contenidos son adicionados y editados de forma similar. Como miembro del sitio se cuenta con una carpeta donde se puede guardar el contenido creado. Permite el trabajo con documentos, imágenes, ficheros, vínculos, tópicos, carpetas y artículos noticiosos. Puede utilizarse como servidor intranet o extranet,

un Sistema de Publicación de documentos y una herramienta de trabajo en grupo para colaborar entre entidades distantes. Realiza el proceso de gestión de objetos en el sitio mediante un sistema de flujo de trabajo por defecto basado en los estados de los objetos y los roles de usuario. (19)

- **Mambo:** Es una aplicación escrita en lenguaje PHP. Permite la creación y mantenimiento de sitios web y portales de manera fácil y dinámica, permitiendo al dueño o administrador de una página web la simplicidad para actualizarla y hacerla accesible a todo tipo de usuarios a través de una variedad de instrumentos. La simplicidad de Mambo radica en que no son necesarios conocimientos técnicos ni especializados para crear, mantener, actualizar o personalizar los contenidos de un sitio Web. (20)
- **Drupal:** Es una plataforma dinámica para la construcción de sitios Web que permite a un individuo o una comunidad de usuarios publicar, manejar y organizar una variedad de contenido. Integra muchas características populares de los CMS, weblogs, herramientas de colaboración y comunidad de discusión, todo en un solo paquete fácil de utilizar. Como software de código abierto desarrollado y mantenido por una comunidad, Drupal es libre para descargarlo de Internet y usarlo. (21)
- **OpenCMS:** Basado en Java y en tecnología XML. Se trata de una aplicación CMS con características tales como entorno de trabajo basado en navegador web, gestión de activos, sistemas de gestión de usuarios y permisos integrados, publicación de contenidos basada en proyectos, gestión de Workflow y tareas, editor WYSIWYG, soporte a la internacionalización, versionado del contenido, mecanismos de plantillas *Java Server Page* (JSP) y XML, soporte para varios idiomas, sistema de ayuda online, publicación dinámica y estática de contenidos, personalización, sistemas de cacheo integrados, mecanismo modular para las extensiones, sistema de programación de trabajos, mecanismo de sincronización, importación y exportación de contenidos, e integración con el servidor de aplicaciones. (22)
- **Wiki:** Si bien es cierto que se puede utilizar un CMS para hacer gestión documental, existen alternativas especializadas que probablemente son más apropiadas. En el caso del wiki, sus aplicaciones están más orientadas a la gestión documental, que exige mucha más organización y ordenación, y son sumamente colaborativas.

Básicamente un wiki es un almacén de páginas web donde todos los usuarios pueden escribir y modificar los contenidos. Una buena cualidad es la de poder incorporar lo documentado, haciéndolo

a la vez más dinámico, y ágil de captar, debido a su capacidad de poder colgar esa misma documentación, adaptándola para una mejor comprensión. El wiki da la posibilidad de ver la información de una forma muy rápida desde cualquier sitio, obtenerla y detectar si el procedimiento es el apropiado, y si no, poder tomar medidas. (23)

- **GIT-DOC:** Es la integración de las soluciones que tradicionalmente se habían desarrollado de forma independiente para satisfacer las diferentes demandas de la gestión documental, que principalmente son: la captura e indexación de documentos, la recuperación de la información, la gestión de contenidos, y la automatización de flujos de trabajo.

Es un sistema de gestión documental integral, completamente adaptable a las necesidades específicas de cada compañía y de cada proyecto de gestión documental, pudiéndose adaptar a cualquier requerimiento, desde la automatización de un proceso de trabajo específico, a la evolución de una organización tradicional hacia una “empresa sin papeles”. Es lo suficientemente flexible y escalable, de tal modo que permite la integración de nuevas funcionalidades de acuerdo a la evolución de las necesidades futuras de la empresa y a los cambios tecnológicos que se produzcan. (24)

- **Extensión de Gestión de Documentos para Microsoft Exchange (DME):** Es un sistema de administración empresarial que tiene la capacidad de almacenar cualquier tipo de documento, desde el momento mismo de su creación hasta que sea necesaria su destrucción, asegurando que toda la información estará disponible durante ese lapso para los integrantes de la organización que requieran tener acceso a ella. Además, DME cuenta con una interfaz totalmente integrada a Windows, se adapta a las aplicaciones de Microsoft Office y también provee una interfaz web, que permite realizar búsquedas, recuperar y guardar documentos on-line para que todos en su empresa dispongan de los datos que necesiten. (25)
- **Subversion (SVN):** Sistema libre de control de versiones y de código fuente abierto, maneja archivos y directorios a través del tiempo. Presenta un repositorio central y este es como un servidor ordinario de ficheros, excepto porque recuerda todos los cambios realizados. Esto le permite recuperar versiones antiguas de sus datos o examinar el historial de cambios de los mismos. Además puede acceder al repositorio a través de la red, lo que le permite ser usado por personas

que se encuentran en distintos ordenadores. A cierto nivel, la capacidad para que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Si se ha hecho un cambio incorrecto a los datos, simplemente se deshace ese cambio. (26)

Todas estas herramientas reúnen un grupo de características las cuales brindan un conjunto de ventajas que demuestran su importancia.

- **Gestión y control efectivo (sencillez, rapidez y ahorro):** De una forma sencilla, la organización tiene acceso instantáneo a toda la documentación necesaria para su actividad de negocio, con las ventajas añadidas de la eliminación de desplazamientos, reducción de tiempo de consultas y tareas de archivo, ahorro de espacio físico, resolución del problema de localización de documentos.
- **Uso racional de los recursos:** La gestión documental facilita que la información se comparta y se aproveche de forma más eficiente y como un recurso colectivo. Como consecuencia, se reducen drásticamente situaciones como la duplicidad de documentos archivados, fotocopias innecesarias y dobles grabaciones de datos. La seguridad y fiabilidad de la información de documentos de gran valor para la organización pueden custodiarse en locales de alta seguridad, garantizando su perfecto estado de conservación mientras que, para el uso diario, se dispone de su réplica electrónica.
- **Productividad y valor añadido:** Una gestión documental, además de ahorro de costes, genera una productividad y valor añadido adicionales, originados por el rápido acceso a la información dentro de la organización y su posterior distribución, sin necesidad de trasladar los documentos.

En la universidad uno de los mecanismos para la GD, aunque no automatizado, lo constituye el EP.

## 1.2. Expediente de Proyecto

El EP se confecciona para registrar de forma documental el proceso de desarrollo de software dentro de los proyectos productivos. En él se gestiona toda la información referente al proceso de elaboración de un software, así como la información referente al propio proyecto. De esta forma se puede planificar y controlar con una mejor calidad y eficiencia el trabajo que se realiza.

## 1.2.1. Expediente de Proyecto de la UCI

Con el objetivo de convertir a la UCI en una Industria de Software de alto prestigio a nivel nacional e internacional se trazó una estrategia para garantizar la calidad de los productos a comercializar. La documentación de todo el proceso es una tarea fundamental para el logro de la calidad del software, por lo que se hizo inminente la creación del EP. Este se confeccionó teniendo en cuenta varios aspectos como son: identificar y revisar la documentación requerida por modelos, estándares y normas de calidad ya existente a nivel internacional: establecer las necesidades propias de los proyectos de UCI y adaptar las plantillas a este entorno, adicionando algunas que por las características propias de la universidad no existen.

Se adaptaron y complementaron algunas plantillas propuestas por la: **IEEE** (The Institute of Electrical and Electronics Engineers- Instituto de Ingenieros Eléctricos y Electrónicos), **NASA** (National Aeronautics and Space Administration- Administración Nacional de Aeronáutica y del Espacio), **ISO** (Organización Internacional para la Estandarización), **DoD** (United States Department of Defense- Departamento de Defensa de Estados Unidos) y **RUP** (Rational Unified Process- Proceso Unificado de Desarrollo del Software); otras fueron confeccionadas basadas en la experiencia personal.

Se elaboró una estrategia para su puesta en marcha que ya está arrojando resultados y se ha logrado estandarizar la documentación generada por los proyectos de la UCI. A pesar de los buenos resultados obtenidos a partir de la implantación del EP, aún existen algunas deficiencias que se deben erradicar. Estas se basan principalmente en la gran cantidad de información redundante que se recoge en él y la baja calidad de la documentación generada.

En vísperas de la importancia del EP se hace necesario mejorar la realización del mismo. En la actualidad el EP se encuentra en su versión 2.0 y pasando por un proceso de mejoras con el objetivo de erradicar todavía deficiencias existentes y que mediante la experiencia alcanzada por los diferentes equipos de desarrollo han sido señaladas.

Para ello se elabora una estructura basada en el uso de Esquemas XML para lograr una mejor organización, de esta forma se gana en calidad y facilidad a la hora de realizar el proceso de GD en el EP. (27)

Con el objetivo de poner en práctica esta nueva estructura surge la necesidad de elaborar una herramienta, que basada en estos esquemas permita visualizar de una forma amigable al usuario toda la información registrada. Además debe gestionar todas las actividades que de forma manual se realizaban en la elaboración del EP, para así ganar en rapidez y calidad de la información.

### 1.3. Análisis crítico de las soluciones existentes

Las herramientas descritas anteriormente realizan y automatizan procesos dentro de la GD, respondiendo a múltiples funcionalidades. Muchas de ellas aplicadas a sectores no relacionados con el mundo de la producción de software.

También existen algunas soluciones informáticas que sin ser automatizadas gestionan y documentan la información relacionada con la producción del software, entre ellas se encuentran la metodología RUP la cual genera artefactos que guían y documentan el proceso de desarrollo del software. Existen estándares como son CMMI<sup>9</sup>, ISO, IEEE, NASA, DoD, que proponen plantillas con la misma finalidad. (28)

Se han establecido algunas normas de calidad por instituciones y empresas para controlar el proceso de GD por ejemplo, Manual de Calidad para control de Documentos y datos aplicado por Softel que constituye una forma de controlar este proceso en dicha empresa. También software como Avila Doc, PAPIRO, BABEL mencionadas anteriormente.

En la UCI no existe una herramienta que automatice la confección de la información registrada en todo el EP, aunque se utilizan algunos sistemas de GD como el *Subversion*. En la Facultad 10 existe un proyecto que trabaja la gestión documental de la producción de software en la UCI. Este consiste en el análisis y diseño de un Habilitador Metodológico para la integración y el desarrollo de aplicaciones en Software Libre.

---

<sup>9</sup> Modelo de capacidad y madurez integrado o sus siglas en Inglés: **C**apability **M**aturity **M**odel **I**ntegration

En la Universidad se han elaborado varios trabajos de diplomas para la automatización de algunos de los flujos de actividades dentro del proceso de desarrollo de software y en específico para roles determinados, esto lo demuestran los trabajos mencionados a continuación:

- *Automatización del proceso de Gestión de la Información de Recursos de la Facultad 3. Rol de Analista. (MIS-007460)*
- *Automatización del proceso de Gestión de la Información de Recursos de la Facultad 3. Rol de diseñador de base de dato. (MIS-007459)*
- *Análisis y Modelado de la Solución Informática para la Gestión Documental de los Registros Públicos.*
- *Sistema de Gestión Documental para el Laboratorio Central de Calidad, 2008 (MIS-007719). Esta tesis propone una investigación para la organización del proceso de GD.*
- *Propuesta de Listas de Chequeo para los Principales Artefactos Generados en el Expediente de Proyecto de Calidad, 2008 (MIS-007552). Es una propuesta para mejorar la estructura de las listas de chequeo.*
- *Herramienta para la Descripción Digital de Documentos de Archivo, 2008 (MIS-007350).*

Pero estas herramientas no responden específicamente a la problemática que se plantea y su trabajo con el EP es con los documentos ya elaborados y con la confección de los mismos.

Una vez analizados los sistemas antes descritos, se llega a la conclusión que debido a que muchos son software propietarios y los costes de mantenimiento y soporte técnico son muy elevados no son factibles para su uso; los que son de licencia gratis y código libre no responden a las necesidades existentes en los proyectos productivos de la Universidad para gestionar y trabajar con la información registrada en el EP. Todo esto afecta de forma directa el proceso de desarrollo de software. Para dar solución al problema antes expuesto surge la idea de desarrollar una aplicación, siendo necesario realizar un estudio de las herramientas y tecnologías óptimas para su elaboración.

## 1.4. Tecnologías y herramientas

### 1.4.1. Lenguajes de programación

En el mundo del desarrollo del software muchos programadores usan lenguajes de programación de alto nivel y orientados a objetos; la decisión del uso de uno u otro para el desarrollo de un software determinado está en las librerías<sup>10</sup> que estos utilizan, en las cuales radica la verdadera riqueza del lenguaje. La elección final de uno u otro lenguaje también dependerá del posible conocimiento que se tenga de la sintaxis del mismo y de su adaptación al medio para el que se quiere programar, ya que es diferente programar para red local, para Windows, Mac o Linux. A continuación se presentarán algunas características, ventajas y desventajas de los principales lenguajes de programación orientados a objetos que más se usan en la actualidad.

### 1.4.2. Lenguaje C++

Bjarne Stroustrup crea una versión experimental denominada "C with Classes" (**C** con clases) hacia 1979, con la intención de proporcionar una herramienta de desarrollo para el kernel Unix en ambientes distribuidos; el objetivo de este nuevo lenguaje era mejorar algunas características del **C** pero manteniendo su basamento en el mismo. Luego, en 1983 adquiere el nombre de "C++" debido a su esencia de **C** y con mejoras como su operador de incremento numeral (++).

C++ es un lenguaje imperativo orientado a objetos, es un súper conjunto de **C**, que nació para añadirle cualidades y características de las que carecía. El resultado es que como su ancestro, sigue muy ligado al hardware subyacente, manteniendo una considerable potencia para programación a bajo nivel, pero se la han añadido elementos que le permiten también un estilo de programación con alto nivel de abstracción.

**C++** no es un lenguaje orientado a objetos puro (en el sentido en que puede serlo Java por ejemplo), se trata simplemente del sucesor de un lenguaje de programación hecho por programadores (de alto nivel) para programadores, al cual le han ido añadiendo todos los elementos que la práctica<sup>11</sup> aconsejaba como necesarios, con independencia de su belleza o purismo conceptual.

---

<sup>10</sup> Conjuntos de código ya realizado que se puede reutilizar en los programas y que ahorran mucho esfuerzo en la programación.

<sup>11</sup> Referente a las buenas prácticas de la programación, la cuales indican consejos para mejorar la comprensión y claridad del código, además de unificar criterios entre distintos programadores. Parten de la base de que la mejor documentación para el programador es el código en sí y establece un lenguaje común entre el programador con su trabajo y el resto de los programadores.

Es versátil, flexible, conciso y muy eficiente. Por muchos años fue el preferido en el desarrollo de aplicaciones. Se ha utilizado para implementar el núcleo de sistemas como Windows y Java.

### 1.4.3. El lenguaje Java

Este es un lenguaje desarrollado por la compañía Sun Microsystem en los años noventa. Está inspirado en **C++** y se proyectó con la finalidad de obtener un producto de pequeñas dimensiones, simple y portátil sobre diferentes plataformas y sistemas operativos ya sea a nivel de código fuente como a nivel de código binario. (29)

**Java** es un lenguaje **orientado a objetos**, eso implica que su concepción es muy próxima a la forma de pensar humana. Es un lenguaje que genera ficheros de clases compiladas, pero estas son en realidad interpretadas por la Máquina Virtual de Java<sup>12</sup>, quien mantiene el control sobre las clases que se estén ejecutando.

Es un lenguaje **multiplataforma**: El mismo código java que funciona en un sistema operativo, funcionará en cualquier otro sistema operativo que tenga instalada la JVM.

Es un lenguaje seguro: La máquina virtual, al ejecutar el código java, realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras, como por ejemplo los punteros.

Su sintaxis ha sido trabajada mejorando la de **C++** logrando mayor sencillez y legibilidad. Presenta mayor robustez al simplificar la gestión de memoria y eliminar las complejidades del manejo explícito de punteros.

Presenta capacidades avanzadas de ejecución *multi-hilo* y proporciona mecanismos de carga dinámica de clases en tiempo de ejecución. Se puede compilar y ejecutar en cualquier plataforma de sistema operativo por ejemplo en Windows, Solaris o Linux gracias a su máquina virtual. Su desarrollo ha sido rápido y exitoso debido a la gran cantidad de grandes empresas colaboradoras que han dado su aporte para enriquecerlo.

La ejecución de programas escritos en Java suele comportarse más lenta que la de aplicaciones de otro lenguaje haciendo un uso voraz de recursos como memoria y procesador. Esto se hace más notorio si la ejecución se basa en cálculos matemáticos complejos o si la aplicación presenta un diseño cargado de componentes visuales.

---

<sup>12</sup> Java Virtual Machine (JVM). Este es un programa que se puede ejecutar sobre varios sistemas operativos y su función es servir de puente o mediador que puede entender tanto el Java bytecode como el sistema sobre el que se quiere ejecutar el mismo. Permitiendo así la portabilidad de programas escritos en este lenguaje.

#### 1.4.4. El lenguaje C#

Actualmente se está utilizando con gran efectividad el nuevo lenguaje **C#** desarrollado por la empresa Microsoft Corporation, como una recopilación de lo mejor de **C++** y **Java**. Es un lenguaje orientado a objetos, tiene una sintaxis muy parecida a **Java** y posee la potencia de **C++**. Tiene algunas ventajas sobre los restantes lenguajes de alto nivel orientados a objetos. El estándar del lenguaje **C#** por excelencia está comprendido en la especificación ECMA-334<sup>13</sup> de la Ecma International<sup>14</sup>.

#### 1.4.5. Plataforma .NET

En 1998 un equipo de trabajo de Microsoft Corporation comenzó a trabajar en el proyecto Próxima Generación de Servicios Windows (en Inglés Next Generation Windows Services) el cual se fusionó con el grupo encargado de liberar Visual Studio 7 con el objetivo de desarrollar un entorno común de ejecución para todos los lenguajes cubiertos por esta herramienta de desarrollo que permitiera a terceras empresas crear lenguajes adaptados al entorno. Luego, en el año 2000 Microsoft publicó este trabajo denominado Microsoft.NET.

La plataforma .NET sirve de mediador entre el programador y las particularidades del sistema operativo para el que se programen las aplicaciones. Una vez terminado el programa, su ejecución se realizaría sobre esta plataforma que entonces mediaría entre él mismo y el sistema operativo. De esta forma, un sistema desarrollado para .NET pudiera ejecutarse en cualquier sistema operativo que tenga instalada una versión de este *framework* como también se le denomina.

Está diseñado para utilizar los servicios web con XML como mecanismo principal de comunicación entre aplicaciones. Posee avanzadas funciones en tiempo de ejecución lo que permite que cualquier aplicación pueda ser convertida en servicios web XML. Permite escribir programas en cualquiera de los lenguajes soportados por la plataforma, incluso utilizar simultáneamente varios lenguajes en un mismo programa. Entre los lenguajes que se han adherido a esta familia se encuentra el ya mencionado **C#**. Agrupa además

---

<sup>13</sup> Estándar internacional que especifica la representación y semántica de programas escritos en C# así como la sintaxis y restricciones de este lenguaje.

<sup>14</sup> Organización internacional que se basa en membrecías de estándares para la comunicación y la información.

al **Microsoft Visual Basic, C++, Java, Pascal**, entre otros conformando un grupo de más de veinte. Para su óptimo aprovechamiento se desarrolló Visual Studio 2005 que ha evolucionado hasta las actuales versiones de Visual Studio 2008.

Acciones como conexiones a bases de datos y la creación de componentes visuales se encuentran empaquetadas en componentes que tienen implementadas todas las funciones necesarias para estos propósitos. Solo basta con arrastrarlos hacia la aplicación y utilizar sus ventajas. Realiza una adecuada gestión de memoria haciendo las aplicaciones más confiables. Utiliza la ejecución en paralelo aportando mayor eficiencia en tiempo de ejecución.

Tiene un componente de seguridad capaz de monitorear las acciones que pueden ser sensibles sobre el sistema operativo controlando quién escribe y ejecuta el código y con qué propósitos lo hace. Inicialmente el *framework* no era gratuito, más tarde Microsoft liberó la versión 2.0 y actualmente se encuentra disponible para su descarga desde el portal web de esa empresa. (30)

#### 1.4.6. Comparación de C# con C++ y Java

A continuación se expone una comparación del C# frente a otros lenguajes, viendo sus principales características y ventajas frente al resto y así algunas desventajas que presentan otros lenguajes con respecto a él que pudiesen atrasar el desarrollo de la aplicación que se desea.

##### 1.4.6.1. Ventajas frente a C/C++ (31)

- Compila a código intermedio independiente del lenguaje en que haya sido escrita la aplicación e independiente de la máquina donde vaya a ejecutarse.
- Realiza la recolección automática de basura.
- Elimina el uso de punteros, en C# no son necesarios aunque permite utilizarlos.
- Posee capacidades de reflexión.
- No hay que preocuparse por archivos de cabecera ".h".
- Es flexible en cuanto al orden de definición de las clases y las funciones.
- No hay necesidad de declarar funciones y clases antes de invocarlas.
- No existen las dependencias circulares.
- Soporta definición de clases dentro de otras.
- No existen funciones ni variables globales, todo pertenece a una clase.

- Todos los valores son inicializados antes de ser usados (automáticamente por defecto, o manualmente desde constructores estáticos).
- No se pueden utilizar valores no booleanos (enteros, coma flotante) para condicionales.
- Es mucho más limpio y menos propenso a errores.

## 1.4.6.2. Ventajas frente a Java

- El rendimiento es, por lo general, mucho mejor.
- CIL<sup>15</sup> está estandarizado, mientras que los *bytecode* de java no lo están.
- Soporta más tipos primitivos (en Inglés *value types*), incluyendo tipos numéricos sin signo.
- Se usan indizadores que permiten acceder a cualquier objeto como si se tratase de un arreglo.
- Compilación condicional.
- Aplicaciones *multi-hilo* más sencillas de manejar.
- Soporta la sobrecarga de operadores, que aunque pueden complicar el desarrollo son opcionales y algunas veces muy útiles.
- Permite el uso de punteros cuando realmente se necesiten, como al acceder a librerías que no se ejecuten sobre la máquina virtual.

## 1.4.6.3. Ventajas frente al C++ y Java

- Concepto formalizado de los métodos *get* y *set*, con lo que se consigue código mucho más legible.
- Gestión de eventos usando delegados mucho más fácil de implementar.

De manera subjetiva es posible decir que este nuevo lenguaje utilizado en aplicaciones como Microsoft Visual Studio simplifica la labor de la programación y minimiza la posibilidad de errores porque todo se ve más sencillo y real. Este lenguaje se desarrolló pensando en las necesidades del programador. Entre las pocas desventajas expuestas por el C# está que es muy orientado a ambientes del sistema operativo Windows. Y no es el más recomendado para sistemas embebidos y núcleos de sistemas.

---

<sup>15</sup> Siglas en el Inglés: **C**ommon **I**ntermediate **L**anguage

## 1.4.7. Selección de un lenguaje de programación

Finalmente se ha comprobado que el lenguaje C++ es muy potente y estable pero complicado para algunos propósitos. Java es muy similar al C# y su sintaxis es amigable pero eventualmente es menos eficiente en la ejecución de programas de escritorio frente a los desarrollados en C#. Además, este último aporta características novedosas que simplifican grandemente las labores de implementación. Se enriquece con la potencia de C++ y la sencillez y modernidad de Java pero mejorando lo que debe ser mejorado. Ha alcanzado gran auge y se utiliza para desarrollar gran cantidad de aplicaciones desde mediano a gran tamaño, obviando esfuerzo de programadores en actividades de rutina. Está debidamente estandarizado. Se usa en múltiples grupos de proyecto en la Facultad 7 de la UCI. Es gratis tanto el C# como la plataforma que lo soporta. Resulta, por tanto, el adecuado para las aspiraciones y necesidades de este trabajo.

## 1.5. Entorno de desarrollo

Un Entorno de Desarrollo Integrado, Integrated Development Environment (IDE) es un programa que agrupa un conjunto de herramientas que viabilizan la labor de los programadores. Los componentes esenciales que los definen son:

- Editor de texto (código).
- Compilador.
- Intérprete.
- Depurador.
- Herramientas de automatización (completamiento de código y navegación rápida dentro del código).
- Sistema de control de versiones.
- Diseñador de interfaces gráficas.
- Agregación de herramientas externas.
- Soporte para varios lenguajes de programación y plataformas de desarrollo.

No todos poseen las anteriores propiedades ni las implementan al mismo nivel, pero esas son las más usuales.

## 1.5.1. Los IDE más utilizados para generar aplicaciones en lenguaje C# para Plataforma .NET

### 1.5.1.1. Entornos Proprietarios

#### **Microsoft Visual Studio:**

Es un IDE para sistemas Windows desarrollado por Microsoft Corporation. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET<sup>16</sup> y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio 2005 permite a los desarrolladores crear sitios, aplicaciones y servicios web y aplicaciones de escritorio en cualquier entorno que soporte la plataforma. Sobre el mismo se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles. (32)

Los programadores de Windows encuentran los nuevos formularios *Windows Forms* intuitivos y muy eficaces. Estos son compatibles con cualquier lenguaje basado en .NET. Con la herencia visual, los programadores pueden simplificar enormemente la creación de aplicaciones basadas en Windows, centralizando en formularios primarios la lógica común y la interfaz de usuario para toda la solución. Utilizando delimitadores y acoplamiento, los programadores pueden generar formularios redimensionables automáticamente, mientras el editor de menús permite crear menús de manera visual directamente desde el *Diseñador de Windows Forms*.

Visual Studio .NET continúa siendo el punto de referencia para la productividad de los programadores. Con un único entorno de programación (IDE) integrado para todos los lenguajes, las organizaciones de programación pueden aprovechar las ventajas de un cuadro de herramientas, un depurador y una ventana de tareas comunes, reduciendo enormemente la curva de aprendizaje del programador y garantizado que siempre puedan elegir el lenguaje más apropiado para sus tareas y conocimientos. Con la función para completar instrucciones de *IntelliSense* y la comprobación automática de errores de sintaxis, Visual Studio .NET informa a los programadores cuándo el código es incorrecto y proporciona el dominio inmediato de las jerarquías de clases y las API. Con el explorador de soluciones, los programadores pueden reutilizar

---

<sup>16</sup> Plataforma de desarrollo de aplicaciones Web desarrollado y distribuido por Microsoft Corporation que brinda la posibilidad de desarrollar aplicaciones Web dinámicas y estáticas así como servicios Web XML.

fácilmente código a través de diferentes proyectos e incluso, generar soluciones multilenguaje que satisfagan con mayor eficacia sus necesidades empresariales.

Gracias al entorno totalmente extensible, se puede disfrutar de las ventajas de una activa comunidad de componentes y complementos de otros fabricantes, con componentes y controles que contribuyen a personalizar y ampliar el entorno de acuerdo con las necesidades.

Los asistentes para aplicaciones, las plantillas de proyecto y los ejemplos de código fuente de Visual Studio .NET permiten crear aplicaciones con rapidez para Windows, la Web y dispositivos con una inversión inicial mínima. La ayuda dinámica y Microsoft Developer Network (MSDN®)<sup>17</sup> proporcionan asistencia basada en la tarea y el lenguaje de programación actuales, garantizado que los programadores sepan aprovechar las oportunidades de la plataforma Microsoft .NET en el lenguaje que hayan elegido. Las macros de Visual Studio, similares a las macros de Microsoft Visual Basic para Aplicaciones en Office, permiten la automatización de tareas rutinarias dentro del IDE, mejorando aún más la productividad global de los programadores de Visual Studio.

La versión Visual Studio 2005 se puede adquirir en las siguientes ediciones:

- Visual Studio 2005 Team System: Para grandes equipos de desarrollo.
- Visual Studio 2005 Professional: Para programadores independientes o pequeños grupos de programadores con múltiples propósitos.
- Visual Studio 2005 Standard y Visual Studio 2005 Express: Ediciones más simples para usuarios que se inician y estudiantes principalmente sin grandes propósitos.

Los precios de compra de las mismas oscilan entre los \$300 dólares y los \$1 400 dólares desde las ediciones Express y Standard hasta la Team System en dependencia del distribuidor, los elementos que incluya el paquete y otros términos a tener en cuenta.(33)

---

<sup>17</sup> Recurso muy útil para el desarrollo mediante herramientas de Microsoft que contiene información técnica sobre programación con código de ejemplo, documentación, artículos técnicos y guías de referencia.

## **C# Builder 2006:**

CodeGear es una filial de la empresa estadounidense desarrolladora de herramientas para la programación Borland Software Corporation. La misma liberó, entre otros paquetes para diferentes lenguajes, el C#Builder 2006 para el lenguaje C#.

Se encuentra disponible en varias versiones especializadas como C# Builder Professional, C# Builder Enterprise, C# Builder Architect. (34)

En general esta opción se ajusta a proyectos con vista a obtener aplicaciones para la plataforma .NET con buenas capacidades de desarrollo empresarial y en la gestión y mantenimiento de proyectos de C# o aquellos que se encuentran en transición de Java y C++ a C#. Entre las características a destacar se encuentran el mapeado relacional de objetos, la persistencia de datos, el modelado UML<sup>18</sup>, la implementación a nivel de diagramas de estados, la gestión de requisitos y la gestión de código fuente. Descargar la aplicación tiene un costo superior a los \$1000 dólares.

### **1.5.1.2. Entornos Libres**

#### **SharpDevelop:**

Es un IDE desarrollado por ICSharpCode Team que soporta el desarrollo de aplicaciones escritas en C#, Visual Basic.NET y BOO<sup>19</sup>. Está desarrollado en C# y es libre. Proporciona todas las características necesarias para un entorno de programación de Windows, entre ellas:

- Autocompletado de código: Esta funcionalidad simplifica el esfuerzo y tiempo de codificación ya que el programador va escribiendo el inicio de las palabras y el editor de texto, ya sea automáticamente o mediante teclas de acceso rápido, despliega un menú contextual con las diferentes palabras que se asocian al lexema que se ha escrito y al contexto en que lo hace.

---

<sup>18</sup> Lenguaje Unificado de Modelado (en Inglés Unified Modeling Language) para la especificación, construcción y visualización de elementos en el desarrollo de software.

<sup>19</sup> Lenguaje de programación orientado a objetos de tipos estáticos muy parecido a Python e integrable con Microsoft .NET y Mono.

- Plantillas de proyectos: Estas plantillas son programas ya comenzados con gran parte de su encabezamiento y primeros elementos especificados y que el programador no tendrá que escribir reduciendo tiempo y esfuerzo de codificación.
- Depurador integrado: Esta es una de las ventajas que más se esperan de un buen IDE ya que permite probar el funcionamiento paso a paso de la sección del programa que se desee. Basta con poner los puntos de ruptura en los sitios escogidos y ordenar la depuración dando la orden a cada paso pudiendo observar los valores que van tomando las variables y pasar por alto llamadas a funciones o insertarse en la ejecución de las mismas. Esto es muy útil para probar algoritmos complejos y da la posibilidad de encontrar un error en mucho menos tiempo que analizando el algoritmo mentalmente.
- Diseñador de formularios: El diseñador de formularios permite conformar la interfaz visual de las aplicaciones de forma rápida y real porque la persona que diseña va viendo la apariencia de su diseño a medida que incorpora y ajusta elementos en la ventana. Paralelo a esto, la herramienta va generando el código fuente correspondiente a cada componente. De esta forma también se ahorra tiempo y esfuerzo.
- Tiene herramientas para *ir a definición*, *encontrar referencias* y *renombrado* que permite desplazarse rápidamente por el código yendo a los puntos deseados con solo dar clic derecho y escoger una de estas opciones. La primera permite desde cualquier invocación a método o variable ir a su definición. La segunda permite lo contrario, desde el nombre de la función o variable, encontrar todas sus invocaciones hechas en el código. La tercera permite que al cambiar el nombre del identificador de alguno de estos elementos, se realicen los cambios automáticamente a todas las ocurrencias de los mismos.
- Integración con herramientas externas: Muy utilizado a la hora de incorporar aplicaciones principalmente las que se dedican a la administración y pruebas de código.
- Posee analizadores para ensamblado.

SharpDevelop es compatible con Visual Studio Express y Visual Studio 2005 por lo que es posible trabajar con proyectos indistintamente en uno u otro ambiente sin cambiar el formato de archivos de proyecto y de código. Su objetivo es brindar una opción para desarrolladores que no tienen la posibilidad de adquirir el Visual Studio.

Soporta además los siguientes lenguajes: HTML, ASP<sup>20</sup>, ASP.NET, Visual Basic Script y XML con sintaxis destacada (escritura de las palabras en diferentes colores en dependencia de su tipo dentro del lenguaje) para todos ellos. Es capaz de convertir código C# a Visual Basic .NET y viceversa y de cualquiera de estos dos hacia BOO. Permite desarrollar proyectos para las siguientes plataformas: Microsoft .NET Framework 1.1 y 2.0, Microsoft .NET Compact Framework 1.0 y 2.0 de la familia .NET; Mono 1.1 y 2.0 de la familia Mono<sup>21</sup>. (35)

Actualmente se encuentra publicado en el sitio de su equipo de desarrollo y libre para descargar bajo la licencia GNU LGPL. Como todo software libre ofrece además su código fuente. Cabe señalar que tiene un alto nivel de compresión ya que el paquete de instalación tiene algo más que 8 MB y el paquete de código fuente tiene aproximadamente 13 MB.

De manera general se puede llegar a la conclusión que SharpDevelop es una alternativa libre y gratuita para realizar programas para la plataforma .NET. Está totalmente basado en el ambiente de trabajo de Visual Studio. Posee las características básicas que se necesitan para caracterizarlo como un moderno entorno de desarrollo de aplicaciones de propósitos múltiples.

### ***MonoDevelop:***

Este es un entorno de desarrollo libre y gratuito desarrollado por Novell y los impulsores de Mono con el objetivo de adaptar las funcionalidades y ventajas de SharpDevelop a Gtk#<sup>22</sup>. Es usado por los desarrolladores de Mono en algunas distribuciones de Linux y de Mac OS.

Sus principales características son:

---

<sup>20</sup> Siglas en Inglés: Active Server Page

<sup>21</sup> Plataforma de desarrollo de código abierto iniciada en 2001 por Ximian e impulsado por Novell que se basa en .NET y permite crear aplicaciones para varios sistemas operativos.

<sup>22</sup> Protegido por la licencia GNU LGPL es un conjunto de bibliotecas y rutinas para desarrollar interfaces gráficas principalmente para sistemas operativos de la familia Linux aunque se puede utilizar en otros.

- Soporta los lenguajes C#, Visual Basic .NET, C/C++, Java y BOO.
- Manejo de clases.
- Completamiento de código.
- Funciones de navegación por el código.
- Diseñador de interfaz gráfica para GTK#.
- Control de versiones integrada con soporte para Subversion.
- Pruebas unitarias integradas con el componente NUnit.
- Posibilidad de crear proyectos ASP.NET.
- Editor y explorador de bases de datos integrado.
- Integración con la herramienta Monodoc para la generación de documentación sobre las clases.
- Compatible con Visual Studio.
- Control de empaquetamiento de código.
- Herramientas de líneas de comando para compilar y administrar proyectos.
- Agregación de herramientas externas para su enriquecimiento.
- Incluye ayuda de la aplicación para el usuario.

Actualmente se encuentra en ampliación y mejoramiento. Su primera versión es MonoDevelop 1.0 publicada bajo licencia GPL el 14 de marzo de 2008 en el sitio del equipo de desarrollo de MonoDevelop (36).

## 1.6. Metodologías de Desarrollo de Software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Es como un libro de recetas de cocina, en el que se van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Actualmente es imprescindible considerar los riesgos a la hora de la creación de un software en cuanto a coste y beneficios antes y después de la concepción del mismo. Una adecuada selección de la metodología servirá para que la confección de la herramienta no parezca complicada, rigurosa y difícil de controlar,

además que se pueda lograr la satisfacción por parte del cliente y que cumpla con el plan de coste y calidad requerido.

En un proyecto de desarrollo de software la metodología define ¿Quién debe hacer qué?, ¿Cuándo y cómo debe hacerlo? Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable, aunque no exista una metodología de software universal.

## 1.6.1. RUP

El Proceso Unificado de Desarrollo de Software (RUP<sup>23</sup>) constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objeto basado íntegramente en Lenguaje Unificado de Modelado (UML) como soporte a la metodología.

*“El RUP es un producto de Rational (IBM<sup>24</sup>). Se caracteriza por ser:*

- *“Iterativo e incremental: La alta complejidad de los sistemas actuales hace que sea factible dividir el proceso de desarrollo en varios mini-proyectos o versiones del producto donde a cada uno de estos se le denomina iteración y pueden o no representar un incremento en el grado de terminación del producto completo.*
- *“Estar centrado en la arquitectura: La arquitectura representa la forma del sistema, la cual va madurando en su interacción con los casos de uso hasta llegar a un equilibrio entre funcionalidad y características técnicas.*
- *“Guiado por los casos de uso: RUP utiliza los casos de uso tanto para especificar los requisitos funcionales del sistema, como para guiar todos los demás pasos de su desarrollo, dígase diseño, implementación y prueba.” (37)*

RUP es considerado como uno de los procesos más generales y completos, es altamente configurable y puede ser aplicado a cualquier tipo de proyecto, incluso en aquellos que no son de desarrollo de software.

El proceso puede describirse en dos dimensiones, o a lo largo de dos ejes:

---

<sup>23</sup> Siglas en Inglés: **R**ational **U**nified **P**rocess

<sup>24</sup> Siglas en Inglés: **I**nternational **B**usiness **M**achines

- El eje horizontal representa tiempo y muestra el aspecto dinámico del proceso, expresado en términos de ciclos, fases, iteraciones, y metas.
- El eje vertical representa el aspecto estático del proceso; como está descrito en términos de actividades, artefactos, trabajadores y flujos de trabajo.

Define cuatro fases: Inicio, Elaboración, Construcción y Transición; nueve flujos de trabajo, seis de Ingeniería (Modelado del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue) y tres de apoyo (Gestión de la Configuración, Gestión de Proyecto y Ambiente).

RUP pretende implementar las mejores prácticas actuales en Ingeniería de Software:

- Desarrollo iterativo del Software.
- Administración de requerimientos.
- Uso de arquitecturas basadas en componentes.
- Modelación visual del software.
- Verificación de la calidad del software.
- Control de cambios.

## 1.6.2. Metodologías Ágiles

Los procesos ágiles de desarrollo de software, conocidos también como metodologías livianas, intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

### ***Programación extrema (XP):***

La programación extrema o *Extreme Programming (XP)* es un enfoque de la Ingeniería de Software formulado por Kent Beck<sup>25</sup>. Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la prevención de cambios. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Crean que ser capaz de adaptarse a los cambios de requisitos en cualquier punto

---

<sup>25</sup> Autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999)

de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software. Mientras que RUP intenta reducir la complejidad del software por medio de estructura y la preparación de las tareas pendientes en función de los objetivos de la fase y actividad actual, XP, como toda metodología ágil, lo intenta por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción.

Las principales características de esta metodología son:

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se obtiene una mejor calidad del código cuando se realiza de esta manera porque es revisado y discutido mientras se está escribiendo.
- Frecuente integración del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nuevas funcionalidades. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenimiento pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir

y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.

- **Simplicidad en el código:** es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

## **Scrum:**

Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años.

Está especialmente indicado para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos:

- El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente.
- La segunda característica importante son las reuniones a lo largo de toda la duración del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

## **Crystal:**

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear<sup>26</sup> (3 a 8 miembros) y Crystal Orange<sup>27</sup> (25 a 50 miembros).

---

<sup>26</sup> Crystal Clear: Color claro.

<sup>27</sup> Crystal Orange: Color naranja

## **1.7. Selección de las tecnologías a usar**

### **1.7.1. Selección de un Entorno de Desarrollo y un lenguaje**

La herramienta a desarrollar es una aplicación típica de escritorio que no posee conexiones a bases de datos. Su función se basa en la del manejo de ficheros XML para su uso en la gestión de la información de las plantillas del expediente de proyecto de la UCI.

De la investigación se llegó a la conclusión de que para estos propósitos era factible utilizar SharpDevelop porque brinda las posibilidades que se requieren para un programa de este tipo. Su descarga desde Internet es gratuita y se instala de forma rápida en la computadora. Es libre y se asemeja bastante al Visual Studio en cuanto al ambiente de trabajo.

De esta forma, una vez analizadas las ventajas y desventajas de todas las posibles combinaciones de lenguaje – plataforma – IDE. Se decidió realizar la implementación mediante el conjunto formado por el lenguaje C#, la plataforma .NET y SharpDevelop como IDE.

### **1.7.2. Metodología a usar**

Muchas son las metodologías que se pudiesen usar en el desarrollo de la aplicación propuesta por este trabajo; las ventajas de usar unas u otras estaría en el historial de trabajo que se logre obtener como resultado final, en el tiempo de desarrollo y en la estimación de riesgo para el control de costo y calidad del producto.

Las metodologías ágiles que han sido descritas anteriormente profundizan mucho en el control de un software en tiempo record cumpliendo con los requisitos establecidos por el cliente y dejando a la opción de una gestión de cambio bastante desarrollada; entre estos, XP es el más usado y de demostrados resultados, su característica de trabajo en pareja es una de las fundamentales porque el código que es ingresado al software es discutido, revisado y probado, obteniendo una mejor calidad en las funciones programadas, aún así posee la dificultad de que es el cliente quien define todo el historial de requisitos y funciones y la aplicación a desarrollar no cuenta con un negocio bien definido por lo cual muchos de los artefactos que XP y otras metodologías tienen no serán posibles de implementar.

RUP es, sin embargo, la indicada entre las metodologías analizadas, pues es la más utilizada para el diseño, implementación y documentación de sistemas informáticos. Es altamente configurable, ya que permite construir solamente los artefactos que se necesiten para el desarrollo de un producto, en este aspecto se puede unir a la fase de construcción y transición del software las mejores características de XP usando así al máximo esta característica de RUP.

En este capítulo se realizó un análisis de las ventajas de desarrollar un buen proceso de GD y se profundizó en las características de los sistemas informáticos que se utilizan a nivel internacional y nacional para este proceso. Se seleccionó dentro de un grupo de tecnologías, herramientas y metodologías las adecuadas para realizar el software que se requiere en la universidad: C# como lenguaje de programación, SharpDevelop como IDE, Framework .NET de Microsoft, RUP como metodología de desarrollo y se determinó como arquitectura la Orientada a Objetos.

## **2. Capítulo II. Características del Sistema**

En este capítulo se describe la herramienta que el presente trabajo de diploma propone y se exponen los procesos del negocio relacionados con el objeto de estudio. Por la poca claridad de los procesos que gestiona dicha herramienta, es necesaria la elaboración de un Modelo de Dominio para que resulte más entendible el contexto en el que se ubica el sistema y así poder construir un sistema con calidad.

### **2.1. Objeto de Automatización**

El proceso de elaboración del Expediente de Proyecto (EP) es muy importante. Es por ello el intento de automatización del mismo, usando la estructura basada en Esquemas XML como estándar para crear, organizar y estructurar el EP. Con esta herramienta se pretende agilizar el proceso de elaboración y confección de los documentos que conforman el EP, así como también se exportará su contenido a formato PDF para su utilización en diferentes sistemas operativos.

### **2.2. Objetivos estratégicos de la UCI como organización**

La UCI, como centro de desarrollo y producción de software persigue como principal objetivo lograr la calidad y eficiencia de los sistemas que se elaboren, para ello consta de varios procesos para el control del desarrollo de software, desde su concepción, obtención del producto, hasta el soporte y mantenimiento y del mismo.

Uno de estos procesos es la GD, que se realiza durante todo la fase de desarrollo; constituyendo el EP uno de los elementos para garantizar la calidad del proceso, por lo que se ha hecho necesario agilizar la elaboración del mismo. Con la creación de una herramienta capaz de gestionar toda la confección del EP se contribuye a alcanzar los objetivos que la Universidad se propone. Se pretende cumplir con el desarrollo de sus procesos y productos, en tiempo record y con la calidad requerida.

## **2.3. Análisis crítico del proceso actual de la situación problemática**

### **2.3.1. Flujo actual de los procesos involucrados en el campo de acción y análisis crítico de la situación problemática**

El proyecto Calidad de la Facultad 7, que también apoya al Laboratorio Central de Calidad de la UCI, desempeña diversas tareas relacionadas con la entrega de productos de alta calidad como resultado del proceso de desarrollo de software, entre ellas se enmarcan la capacitación a los proyectos productivos en el llenado y confección del EP como parte del proceso de GD.

Actualmente las personas que integran dichos proyectos no tienen la experiencia necesaria para elaborar con calidad el historial de desarrollo de un producto, afectando su entrega en tiempo; siendo este un factor fundamental e imprescindible para la entrega del software.

La confección de los diferentes artefactos que componen el EP, en su mayoría, son llenados por el documentador de cada proyecto y otros, por el resto de los roles que conforman los proyectos productivos, pero igual realizándose la elaboración de estos de forma manual y en muchas ocasiones alterándose la uniformidad en su estructura. La Universidad tiene definido patrones de arquitectura estándares para todos los software y su documentación, aunque estos no siempre se cumplen, en ocasiones se omite el llenado de diversos documentos y/o formularios por ser demasiado complejo o porque requieren mucho tiempo para su confección, resultando estos artefactos muy importantes tanto para el mismo proyecto como para el grupo de control de la calidad.

Como resultado, algunos de los problemas que afectan la calidad de este proceso son:

- Existencia de información redundante en las diferentes plantillas.
- La no existencia de un mecanismo que evite que se altere la uniformidad en la estructura de las plantillas.
- Complejidad en la estructura de algunas de las plantillas que dificultan su llenado.
- La omisión de datos de la estructura de las plantillas (Ej.: autores, fecha, módulo, versión).

Con el uso de la herramienta que se propone en este trabajo se garantiza la automatización del proceso y la solución real de estos problemas.

## 2.4. Descripción de términos comunes

Se denominará **Usuario** a cualquier persona que trabaje con la aplicación, sin tener en cuenta la categoría.

Se le denomina **Exportar** a la representación del contenido de un documento XML a un formato determinado. (Ej.: .pdf)

Se le denomina **Plantilla** a un artefacto del EP.

Se le denomina **Documento XML** a una plantilla determinada del EP en la cual se guardarán los datos de dicha plantilla.

Se le denomina **Importar** al proceso de carga de un documento XML para su visualización.

## 2.5. Modelo de Dominio

El modelo de dominio muestra las clases conceptuales significativas en el dominio del problema, captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará la herramienta. Es considerado un subconjunto del llamado modelo de objetos del negocio.

Durante el desarrollo de la herramienta no se pudo contactar procesos bien definidos en el entorno del negocio. Se hizo difícil determinar los elementos más importantes del sistema y sus interconexiones, así como el establecimiento de las reglas de funcionamiento. Sin embargo se pueden identificar personas, eventos y objetos involucrados en ese entorno que no está bien delimitado, por lo que se hizo necesario un modelado del dominio perteneciente a la solución.

*Esto “ayuda a los usuarios, a utilizar un vocabulario común para poder entender el contexto en que se ubica el sistema. Y para capturar correctamente los requisitos y poder construir un sistema correcto se necesita tener un firme conocimiento del funcionamiento del objeto de estudio. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.” (38)*

2.5.1. Modelo de Dominio

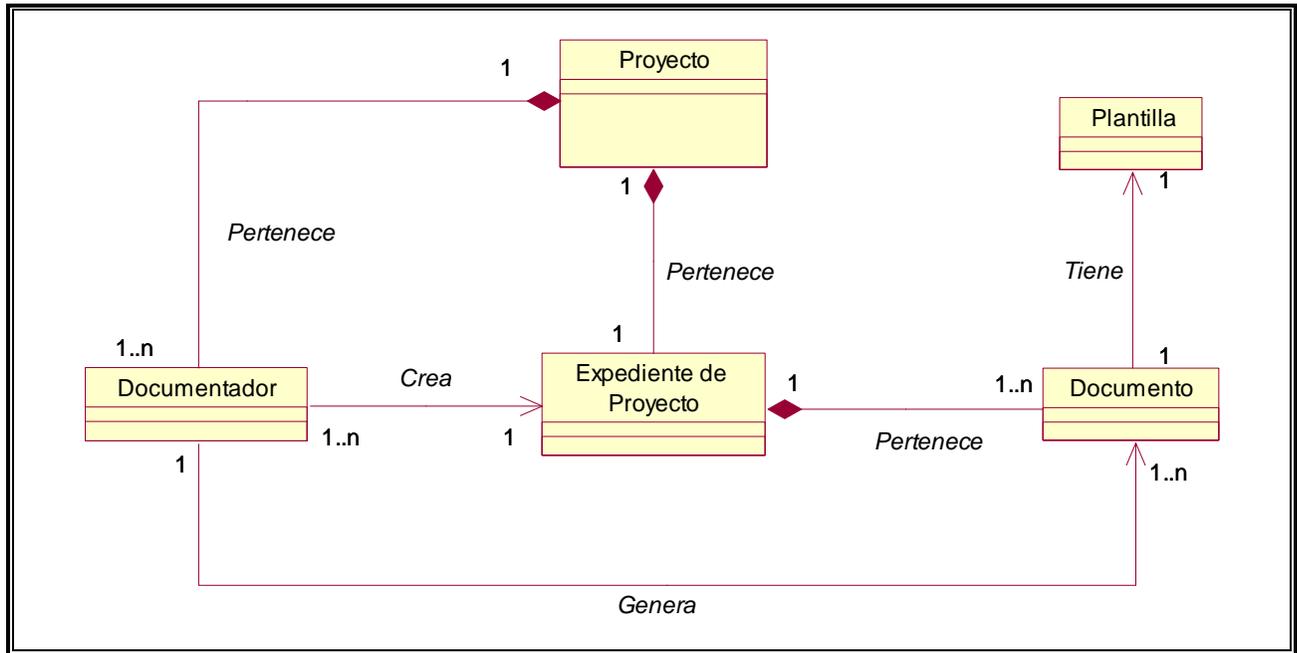


Diagrama 1 Modelo del Dominio

2.5.2. Modelo de Objetos

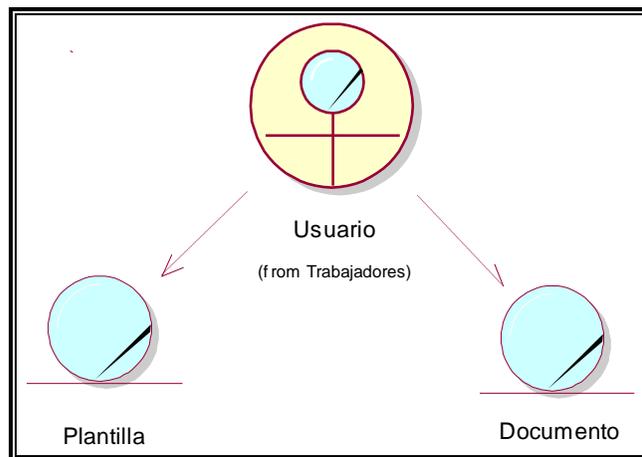


Diagrama 2 Modelo de Objetos

## 2.6. Propuesta del Sistema

*Descripción general de la propuesta de sistema, ¿Cómo debe funcionar?*

Este sistema informático debe dar solución a la gestión de la información en el EP para los proyectos productivos. Dicha aplicación debe funcionar de la siguiente forma:

1. En la aplicación se registrará toda la información referente al EP.
2. El usuario llena la información referente al proyecto que creará, para después elaborar su EP, si ya está creado cargará toda la información referente al proyecto.
3. Si lo que se desea es crear el EP, el usuario selecciona las plantillas que desea utilizar y después pasa a llenar contenido correspondiente.
4. El sistema registrará toda la información en los documentos XML y generará los Esquemas XML para cada plantilla y de esta manera poder comprobar su autenticidad en el futuro.
5. Una vez llenados los datos correspondientes se exportará la plantilla al formato PDF.
6. Se registra la información en un documento XML para futuros cambios o revisiones.
7. A petición del usuario el sistema verificará los documentos XML contra sus Esquemas para verificar su integridad.

## 2.7. Especificación de Requisitos

Es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluyen un conjunto de casos de uso que describen todas las interacciones que tendrán los usuarios con el software. Los casos de uso también son conocidos como requisitos funcionales. Además de los casos de uso y la especificación de requisitos del sistema, también contiene requisitos no funcionales (o complementarios). Los requisitos no funcionales son requisitos que imponen restricciones en el diseño o la implementación (Como por ejemplo restricciones en el diseño o estándares de calidad).

### 2.7.1. Requisitos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, especifican acciones que el sistema debe ser capaz de realizar. Ellos deben de ser comprensibles por los clientes,

usuarios y desarrolladores, deben tener una sola interpretación y estar definidos en forma medible y verificable.

Es por ello que una vez conocido los conceptos que rodean al objeto de estudio, se analiza ¿Qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio del trabajo de diploma?, enumerándose a través de requerimientos funcionales las instrucciones que el sistema deberá ser capaz de efectuar. Dentro de ellos se incluyen las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema y las condiciones extremas a determinar por el sistema.

A continuación se muestran los requisitos que debe cumplir el sistema:

### **RF1 – Gestionar Proyectos.**

- **RF1.1-** Crear nuevo proyecto.
- **RF1.2-** Visualizar interfaz para la Información.
- **RF1.3-** Gestionar Módulos.
- **RF1.4-** Editar información del proyecto.
- **RF1.5-** Eliminar proyecto.
- **RF1.6-** Abrir Expediente del Proyecto.
- **RF1.7-** Guardar Información del proyecto.
- **RF1.8-** Agregar miembros del Proyecto.

### **RF2 – Validación de XML contra su Esquema XML.**

### **RF3 – Gestionar Documentos.**

- **RF3.1 –** Cargar imágenes.
- **RF3.2 –** Salvar imágenes.
- **RF3.3 –** Editar Información.
- **RF3.4 –** Guardar información en formato XML.
- **RF3.5 –** Generar información en formato PDF.
- **RF3.6 –** Eliminar información.

### **RF4 – Gestionar Información en Documentos XML.**

- **RF4.1 –** Editar Información.
- **RF4.2 –** Eliminar Información.

### **RF5 – Generar PDF.**

## 2.7.2. Requisitos No Funcionales

Con el propósito de responder las necesidades de los proyectos productivos de la UCI, se definió un conjunto de propiedades o cualidades que debe cumplir la aplicación, las cuales se describen a continuación.

### Apariencia o Interfaz Externa:

- **RNF1-** Herramienta Sencilla.
  - **RNF1.1-** Amigable al usuario y fácil de emplear.
  - **RNF1.2-** Funcionalidades bastantes explícitas.
- **RNF2-** Colores Relajantes.
- **RNF3-** No habrá sobrecarga de imágenes.

### Usabilidad:

- **RNF4** -La herramienta estará orientada al responsable de ocupar el rol de Documentador, el cual debe conocer el cómo se confecciona cada una de las plantillas para contener la información del Expediente de Proyecto.
- **RNF5** - La Interfaz es de un flujo sencillo, típico de herramientas de escritorio.
- **RNF6-** Será útil para manejar la información y elaboración del EP.

### Soporte:

- **RNF7-** Se requiere de la información a registrar.
- **RNF8-** Contará con una Ayuda que permitirá el aprendizaje sobre el uso de la Aplicación y confección de elementos de las plantillas que sea de difícil entendimiento.

### Portabilidad y software:

- **RNF9-** Estará soportado sobre sistemas con Framework 3.5.
  - **RNF9.1-** Windows 2000 Professional con SP4 y Microsoft .Net Framework 2.2.
  - **RNF9.2-** Windows XP Professional con SP2 y Microsoft .Net Framework 2.2
  - **RNF9.3-** Windows XP Home Edition con SP2 y Microsoft .Net Framework 2.2.
- **RNF10-** Estará soportado sobre sistemas GNU con Mono 1.v o superior.

## Seguridad:

- **RNF11-** El sistema debe proteger la integridad de la información que se maneja.
- **RNF12-** El sistema no debe presentar fallos y en casos de alguno debe garantizar que las pérdidas de información sean mínimas.

## Políticos – culturales:

- **RNF13-** La herramienta sólo podrá ser utilizada dentro de la UCI.

## Hardware:

- **RNF14-** Rendimiento mínimo del hardware.
  - Procesador 600 MHz o superior.
  - 128 MB de memoria RAM.
  - RNF Monitor VGA o superior.
  - Ratón Microsoft o compatible.

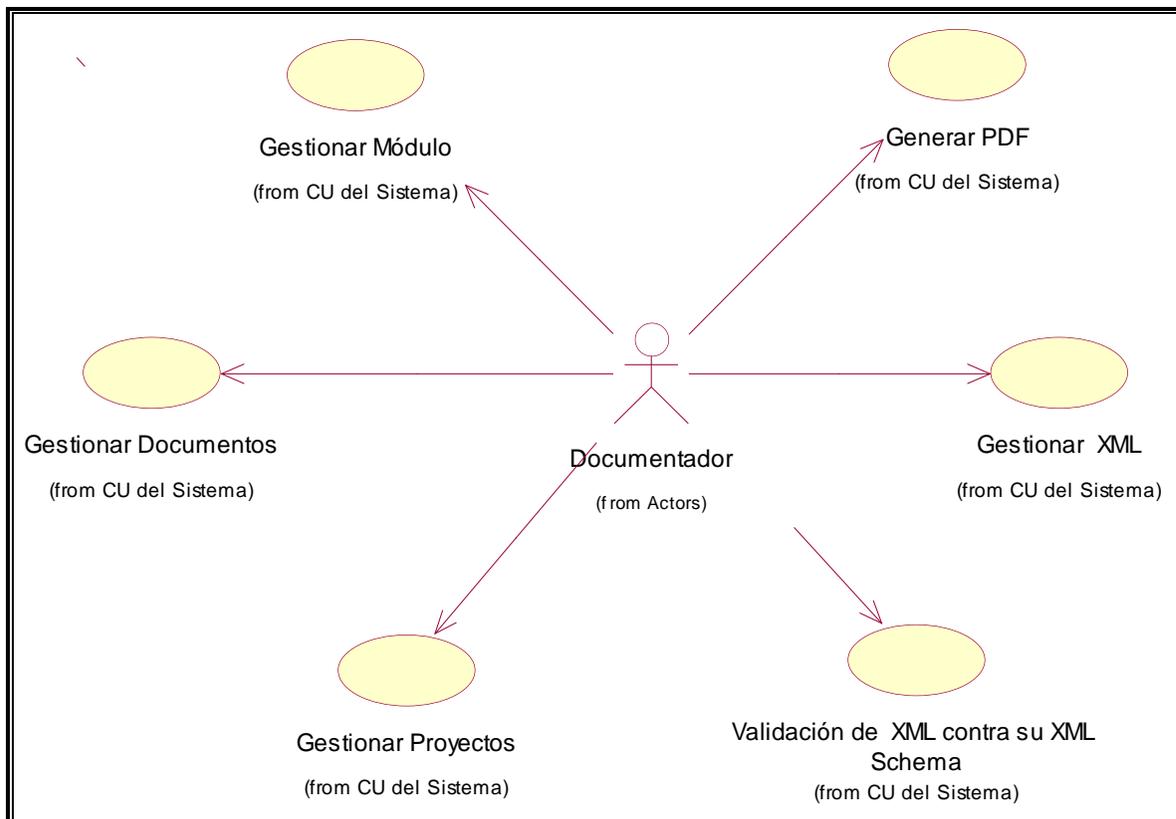
## Restricciones para el diseño de la implementación:

- **RNF15-** Utilizar los estándares de diseño establecidos.
- **RNF16-** Para la implementación de la herramienta se deberá utilizar SharpDevelop con lenguaje de programación C#.

## 2.8. Definición de los Casos de Usos del Sistema (CUS)

- **CUS 1** Gestionar Proyectos.
  - **1.1** Nuevo Proyecto.
  - **1.2** Abrir Proyecto.
  - **1.3** Modificar Proyecto.
  - **1.4** Eliminar Proyecto.
  
- **CUS 2** Gestionar Módulos.
  - **2.1** Agregar Módulo.
  - **2.2** Eliminar Módulo.

- **CUS 3** Validación de XML contra su Esquema XML.
- **CUS 4** Gestionar Documentos.
  - 4.1 Confeccionar Información.
  - 4.2 Editar Información.
  - 4.3 Eliminar Información.
- **CUS 5** Gestionar Información en Documentos XML.
  - 5.1 Editar información.
  - 5.2 Eliminar información.
- **CUS 6** Generar PDF.



**Diagrama 3** Casos de Usos del Sistema

**Autores:**

**Definición de los actores**

Actores	Justificación
Documentador	Es el encargado del manejo de toda la información documental dentro de un proyecto productivo.

**Especificación de los CU**

<b>CUS 1</b>	Gestionar Proyectos.
<b>Actor</b>	Documentador
<b>Referencia</b>	RF 1, RF 1.1, RF 1.2, RF 1.3 ,RF 1.4, RF 1.5, RF 1.6, RF 1.7, RF 1.8, RF 2, RF 4, RF 5
<b>Descripción</b>	El actor podrá crear un nuevo proyecto, el cual almacenará la información referente al proyecto, así como la edición o eliminación de uno existente eliminando todos los archivos que existan de registro del mismo. También se permitirá cerrar el proyecto con el que se trabaja y cargar uno otro. Se visualizará un explorador con el proyecto creado y los módulos que lo conforman, mostrando también la estructura organizativa del contenido del EP.

<b>CUS 2</b>	Gestionar Módulos.
<b>Actor</b>	Documentador
<b>Referencia</b>	RF 1, RF 1.3 , RF 4, RF 4.1
<b>Descripción</b>	El actor podrá adicionar y eliminar los módulos al expediente del proyecto así como los registros que se crean para cada uno.

<b>CUS 3</b>	Validación de XML contra su XML.
--------------	----------------------------------

<b>Actor</b>	Documentador
<b>Referencia</b>	RF 2
<b>Descripción</b>	El actor validará la autenticidad del documento XML con su Esquema correspondiente mostrando los errores que pueda este contener para su futuro uso.

<b>CUS 4</b>	Gestionar Documentos.
<b>Actor</b>	Documentador
<b>Referencia</b>	RF 3, RF 3.1, RF 3.2, RF 3.3, RF 3.4, RF 3.5, RF 3.6, RF 4, RF 4.1, RF 4.2, RF 5
<b>Descripción</b>	El actor podrá gestionar la información de los documentos que se están elaborando, podrá elegir entre las opciones de editar un documento previamente guardado, cargar imágenes para insertarlas al documento, guardar la información y generar un entregable en formato PDF.

<b>CUS 5</b>	Gestionar Información en Documentos XML.
<b>Actor</b>	Documentador
<b>Referencia</b>	RF 4, RF4.1, RF4.2
<b>Descripción</b>	La información que se encuentra en los documentos XML podrá ser editada y eliminada siempre que el actor lo requiera sin poder variar su estructura.

<b>CUS 6</b>	Generar PDF.
<b>Actor</b>	Documentador
<b>Referencia</b>	RF5
<b>Descripción</b>	El actor podrá siempre que lo desee obtener un documento en formato PDF cada vez que quiera, para ir conformando los documentos del expediente de proyecto listos para su revisión y entrega.

## 2.9. Casos de Usos Extendidos

Caso de uso	
<b>CUS 1</b>	Gestionar Proyectos.
<b>Propósito</b>	Gestionar la creación de un nuevo proyecto, eliminación, trabajo con uno ya existente.
<b>Actores:</b> Documentador	
<b>Resumen:</b> El actor podrá crear un nuevo proyecto, el cual almacenará la información referente al proyecto, así como la edición o eliminación de uno existente eliminando todos los archivos que existan de registro del mismo. También se permitirá cerrar el proyecto con el que se trabaja y cargar otro. Se visualizará un explorador con el proyecto creado y los módulos que lo conforman, mostrando también la estructura organizativa del contenido del EP.	
<b>Referencias</b>	RF 1, RF 1.1, RF 1.2, RF 1.3 ,RF 1.4, RF 1.5, RF 1.6, RF 1.7, RF 2, RF 4, RF 5
Acción del actor	Respuesta del sistema
1. El actor accede a la barra de menú:	2. La interfaz visual del sistema muestra una barra menú y de herramientas permitiendo realizar las diferentes opciones: <ul style="list-style-type: none"> <li>a) Crear un nuevo proyecto. Ver <b>Sección “Nuevo proyecto”</b>.</li> <li>b) Abrir proyecto existente. Ver <b>Sección “Abrir proyecto”</b>.</li> <li>c) Cerrar la información general del proyecto. Ver <b>Sección “Cerrar Proyecto”</b>.</li> </ul>
3. El actor marca el proyecto y selecciona la opción deseada.	4. La interfaz visual del sistema muestra una barra menú y de herramientas permitiendo realizar las diferentes opciones: <ul style="list-style-type: none"> <li>d) Eliminar un proyecto. Ver <b>Sección “Eliminar Proyecto”</b></li> <li>e) Modificar proyecto existente. Ver <b>Sección “Modificar”</b></li> </ul>
<b>Sección “Nuevo proyecto”</b>	
Acción del actor	Respuesta del sistema

<p>3. El actor selecciona la opción de <b>Nuevo Proyecto</b> en la barra de herramientas o en la barra de menú.</p>	<p>4. El sistema muestra una vista con la información a llenar para la creación del nuevo proyecto.</p> <ul style="list-style-type: none"> <li>• Nombre del Proyecto.</li> <li>• Líder del proyecto.</li> <li>• Documentador</li> <li>• Módulos.</li> <li>• Versión.</li> </ul>
<p>5. El actor llena los campos correspondientes y seleccionar la opción <b>Guardar</b>.</p>	<p>6. El sistema verifica los datos y registra la información.</p>
<p><b>Flujo Alterno</b></p>	
<p><b>Acción del actor</b></p>	<p><b>Respuesta del sistema</b></p>
	<p>6. El sistema muestra un mensaje de error con los campos que deben ser arreglados.</p>
<p>7. El actor modifica los campos correspondientes y seleccionar la opción <b>Guardar</b>.</p>	<p>8. El sistema verifica los datos y registra la información.</p>
<p><b>Sección “Abrir proyecto”</b></p>	
<p><b>Acción del actor</b></p>	<p><b>Respuesta del sistema</b></p>
<p>3. El actor selecciona la opción de <b>Abrir Proyecto</b> en la barra de herramientas o en la barra de menú.</p>	<p>4. El sistema muestra una interfaz para seleccionar el proyecto que desea abrir.</p>
<p>5. El Actor selecciona el proyecto a cargar.</p>	<p>6. El sistema carga todos los archivos existentes del proyecto.</p>
<p><b>Sección “Cerrar proyecto”</b></p>	
<p><b>Acción del actor</b></p>	<p><b>Respuesta del sistema</b></p>

3. El actor selecciona la opción de <b>Cerrar Proyecto</b> en la barra de herramientas.	4. El Sistema cierra el proyecto que esté abierto.
<b>Sección “Modificar”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
5. El actor selecciona la opción de <b>Modificar</b> con clic derecho sobre el proyecto abierto.	6. El sistema muestra una vista con la información a modificar del proyecto. <ul style="list-style-type: none"> <li>• Nombre del Proyecto.</li> <li>• Líder del proyecto.</li> <li>• Documentador</li> <li>• Módulos.</li> <li>• Versión.</li> </ul>
7. El actor cambia los campos y seleccionar la opción <b>Guardar</b> .	8. El sistema verifica los datos y registra la información.
<b>Flujo Alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	8. El sistema muestra un mensaje de error con los campos que deben ser arreglados.
9. El actor modifica los campos correspondientes y seleccionar la opción <b>Guardar</b> .	10. El sistema verifica los datos y registra la información.
<b>Sección “Eliminar proyecto”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
5. El actor selecciona la opción de <b>Eliminar Proyecto</b> con clic derecho sobre el proyecto.	6. El sistema muestra un mensaje de aviso para reafirmar la acción.

7. El actor selecciona la opción <b>Eliminar</b> .	8. El sistema elimina el proyecto con todos los archivos existentes de su expediente.
<b>Flujo Alternativo</b>	
7. El actor selecciona la opción <b>Cancelar</b> .	8. El sistema cancela la operación.

<b>Caso de uso</b>	
<b>CUS 2</b>	Gestionar Módulos.
<b>Propósito</b>	Poder agregar un nuevo módulo al proyecto o eliminarlo.
<b>Actores:</b> Documentador	
<b>Resumen:</b> El actor podrá adicionar y eliminar los módulos al expediente del proyecto así como los registros que se crean para cada uno.	
<b>Referencias</b>	RF 1, RF 1.3 , RF 4, RF 4.1
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El actor abre la ventana de Modificar Proyecto o Nuevo Proyecto donde está la información de los módulos.	2. El sistema muestra la Interfaz dando las opciones de:  Agregar un nuevo módulo (+).  Eliminar Módulo (-).
<b>Sección "Agregar (+)"</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
3. Se agrega la información del módulo y acepta.	4. El sistema verifica que no exista y guarda la información, agrega el módulo y crea un directorio donde se archivará la información del Expediente del mismo.
<b>Flujo Alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

	4. El sistema verifica que no exista y si ocurre envía un mensaje indicando que ya existe.
5. El usuario rectifica el nombre.	6. El sistema agrega el nuevo módulo.
<b>Sección “Eliminar (-)”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
3. Selecciona eliminar el módulo.	4. El sistema elimina todos los archivos y el directorio referentes a ese módulo.
<b>El Caso de Uso termina</b>	

<b>Caso de uso</b>	
<b>CUS 3</b>	Validación de XML contra su XML.
<b>Propósito</b>	Actualizar el estándar de las plantillas del EP.
<b>Actores:</b> Documentador	
<b>Resumen:</b> El actor validará la autenticidad del documento XML con su Esquema correspondiente.	
<b>Referencias</b>	RF 2
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El actor selecciona en el submenú de una plantilla la opción de “Validar Esquema”.	2. El Sistema muestra una interfaz para la comprobación del documento XML correspondiente a la plantilla contra su esquema.
3. El actor selecciona la opción en la barra de botones de Validar Esquema.	4. El sistema verifica la existencia del XML y el Esquema seleccionado y muestra el resultado de la validación. En caso de error el sistema muestra donde está el error.
<b>Flujo Alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

	3. El sistema verifica la existencia del XML y el Esquema seleccionado mostrando una advertencia.
<b>El Caso de Uso termina</b>	

Caso de uso	
<b>CUS 5</b>	Gestionar Información en Documentos XML.
<b>Propósito</b>	Editar y eliminar siempre que el actor lo requiera, la información que se encuentra en los documentos XML.
<b>Actores:</b> Documentador	
<b>Resumen:</b> Se actualiza la información correspondiente a alguna plantilla del expediente del proyecto, la información es guardada en documentos XML y esta información debe ser editada y rectificada, al actor se le debe dar la posibilidad de borrar toda la información de estos XML y generar uno nuevo. Un documento XML es creado solo cuando el actor comienza a insertar datos correspondientes a los Esquemas XML.	
<b>Referencias</b>	RF 2, RF2.1, RF2.2
<b>Sección “Editar Información”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El actor selecciona nuevamente la plantilla a editar y selecciona la sección que desea modificar.	2. El sistema abre el formulario correspondiente a la sección seleccionada.
3. El actor modifica la plantilla y salva los cambios.	4. El sistema guarda los cambios realizados.
<b>Sección “Eliminar XML”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El actor selecciona la plantilla que desea eliminar y va a la opción eliminar.	2. El sistema le muestra un mensaje de advertencia sobre la acción: <ul style="list-style-type: none"> <li>• SI</li> <li>• NO</li> </ul>

3. El actor selecciona la afirmación sobre la acción a realizar (SI).	4. El sistema elimina el documento XML correspondiente a la plantilla del expediente de proyecto seleccionado.
<b>FLUJO ALTERNO</b>	
3. El actor selecciona la negación sobre la acción a realizar (NO).	4. El sistema cancela la acción y retorna a la ventana principal.

<b>Caso de uso</b>	
<b>CUS 4</b>	<b>Gestionar Documentos.</b>
<b>Propósito</b>	Garantiza el registro de la información en las plantillas, eliminación o su edición.
<b>Actores:</b> Documentador	
<b>Resumen:</b> El actor podrá gestionar la información de los documentos que se están elaborando, podrá elegir entre las opciones de editar un documento previamente guardado, cargar imágenes para insertarlas al documento, guardar la información y generar un entregable en formato PDF.	
<b>Referencias</b>	RF 3, RF 3.1, RF 3.2, RF 3.3, RF 3.4, RF 3.5, RF 3.6, RF 4, RF 4.1, RF 4.2, RF 5
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El actor selecciona el menú desplegable del proyecto y marca el documento con el que desea trabajar, a través de la opción clic derecho escoge la opción deseada.	2. El sistema muestra un menú con las opciones de: <ul style="list-style-type: none"> <li>• <b>Editar Plantilla.</b></li> <li>• <b>Eliminar XML</b> (opción de eliminar el documento).</li> </ul>
<b>Sección "Editar Plantilla"</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

3. El actor selecciona Editar Plantilla y accede a la información de la plantilla.	4. El sistema muestra el contenido de la plantilla seleccionada y la Opción de:  <b>Guardar.</b>
5. El actor selecciona entre introducir el contenido o modificar el existente.	6. El sistema verifica los datos y guarda, si los datos están incorrectos Ver Flujo Alternativo.
<b>Flujo Alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	6. El sistema verifica los datos y muestra un mensaje de error en el campo incorrecto.
7. El actor modifica los campos correspondientes y selecciona la opción <b>Guardar.</b>	8. El sistema si están correctos los datos registra la información.
<b>Sección "Eliminar XML"</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
3. El actor selecciona Eliminar XML.	4. El sistema muestra un mensaje de confirmación.
5. El actor confirma la acción o cancela la operación.	6. El sistema ejecuta la acción.
<b>El Caso de Uso termina</b>	

<b>Caso de uso</b>	
<b>CUS 6</b>	Generar PDF.
<b>Propósito</b>	Generar la plantilla elaborada en un documento formato PDF.

<b>Actores:</b> Documentador	
<b>Resumen:</b> El actor podrá siempre que lo desee obtener un documento en formato PDF para ir conformando los documentos del expediente de proyecto listos para su revisión y entrega.	
<b>Referencias</b>	RF5
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El actor selecciona la opción: <b>PDF o Ver PDF.</b>	2. El sistema muestra una ventana para seleccionar donde desea guardar el documento generado.  Si lo que desea es ver el PDF el sistema automáticamente muestra el documento sin guardarlo todavía.
3. El actor selecciona el nombre y el URL.	4. El sistema guarda el documento generado.
<b>El Caso de uso termina</b>	

## 2.10. Arquitectura del Sistema

La arquitectura de los Sistemas Informáticos (SI) es una descripción de los subsistemas, componentes y las relaciones entre ellos. Determina:

- La organización estructural del SI.
- La selección de los elementos estructurales.
- Las interfaces entre ellos.
- El comportamiento de los componentes.

Existen patrones de arquitecturas definidos para diferentes tipos de aplicaciones, ejemplo:

### Llamada y retorno:

Estilo más usado en los grandes SI

- Descomposición modular jerárquica.
- Orientado a objetos.

- En capas.

### **Centrado en los datos:**

Permite la manipulación compartida de datos

- Repositorio
- Pizarra

### **Flujo de datos:**

Permite la transformación incremental de los datos

- Batch secuencial.
- Tubos y filtros.

### **Modelo – vista – controlador:**

Para sistemas interactivos.

Por las características que reúne el software desarrollado se decide utilizar el Patrón Orientado a Objeto (OO), ya que este ofrece ventajas como:

### **Orientación a objetos**

- Clases encapsulan datos y métodos.
- La comunicación entre componentes se realiza mediante la invocación de servicios ofertados por ellos.
- Facilita la modularidad y la reusabilidad.

## **2.11. Características finales del sistema**

El sistema será una herramienta de fácil uso y todo el proceso de elaboración del EP será guiado por un manual de ayuda bastante claro. La mayor cantidad de los problemas que afectan el llenado de este expediente serán eliminados con el uso de la aplicación.

En el capítulo se abordaron los objetivos específicos de la Universidad en función de la calidad del proceso de GD. Así como detalles de la situación problemática y del entorno donde se desarrolla el llenado del EP. Todo lo anterior permitió definir la primera propuesta del sistema a confeccionar, los requisitos y características del mismo y se desarrollaron los CUS que describen las principales funcionalidades.

### **3. Capítulo III. Análisis y Diseño**

En este capítulo se desarrolla el flujo de análisis y diseño, es uno de los flujos de trabajo del RUP, metodología empleada en la modelación del sistema. Estos dos procesos se analizan de forma independiente aunque están relacionados.

Se realiza el análisis del sistema, modelando su principal diagrama (Diagrama de Análisis) y el diagrama de clases del diseño que es uno de los artefactos principales del flujo de trabajo del RUP, se tratan los principios del diseño de la aplicación, donde se desarrolla el Diagrama de Clases de Diseño.

#### **3.1. Flujos de Trabajo de Análisis y Diseño**

El objetivo principal de esta disciplina es transformar los requerimientos a una especificación que describa cómo implementar el sistema. El análisis fundamentalmente consiste en obtener una visión que se preocupa de ver que hace el sistema de software a desarrollar, por tal motivo este se interesa en los requerimientos funcionales. Por otro lado, el diseño es un refinamiento que toma en cuenta los requerimientos no funcionales, por lo cual se centra en cómo el sistema cumple sus objetivos.

Los objetivos específicos del análisis y diseño son:

1. Transformar los requerimientos al diseño del futuro sistema.
2. Desarrollar una arquitectura para el sistema.
3. Adaptar el diseño para que sea consistente con el entorno de implementación.

Al principio de la fase de elaboración hay que definir una arquitectura candidata: crear un esquema inicial de la arquitectura del sistema, identificar clases de análisis y actualizar las realizaciones de los casos de uso con las interacciones de las clases de análisis. Durante la fase de elaboración se va refinando esta arquitectura hasta llegar a su forma definitiva. En cada iteración hay que analizar el comportamiento para diseñar componentes.

## 3.2. Modelo del 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.

Este modelo de análisis no es un diagrama final que describe todos los posibles conceptos y sus relaciones, es un primer intento por definir los conceptos claves que describen el sistema. Este artefacto es opcional, pero también tiene a su vez la propiedad de ser temporal. Su utilidad radica en que permite una apreciación global conceptual del sistema.

A diferencia del modelo de casos de uso que captura la funcionalidad del sistema, el modelo de análisis da forma a la arquitectura para soportar las funcionalidades que en el anterior modelo se expresa.

Para representar los diagramas del modelo de análisis se pueden emplear diferentes diagramas de UML tales como:

1. Diagramas de Clase.
2. Diagramas de Secuencia.
3. Diagramas de Colaboración.

### 3.2.1. Diagrama de clases del análisis

El **diagrama de clases** es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

- Las clases Interfaz modelan la interacción entre el sistema y sus actores.
- Las clases Controladoras coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.
- Las clases Entidad modelan información que posee larga vida y que es a menudo persistente.

Se presentarán los diagramas de clases del análisis de los casos de uso: Gestionar Documentos y Gestionar Proyectos.

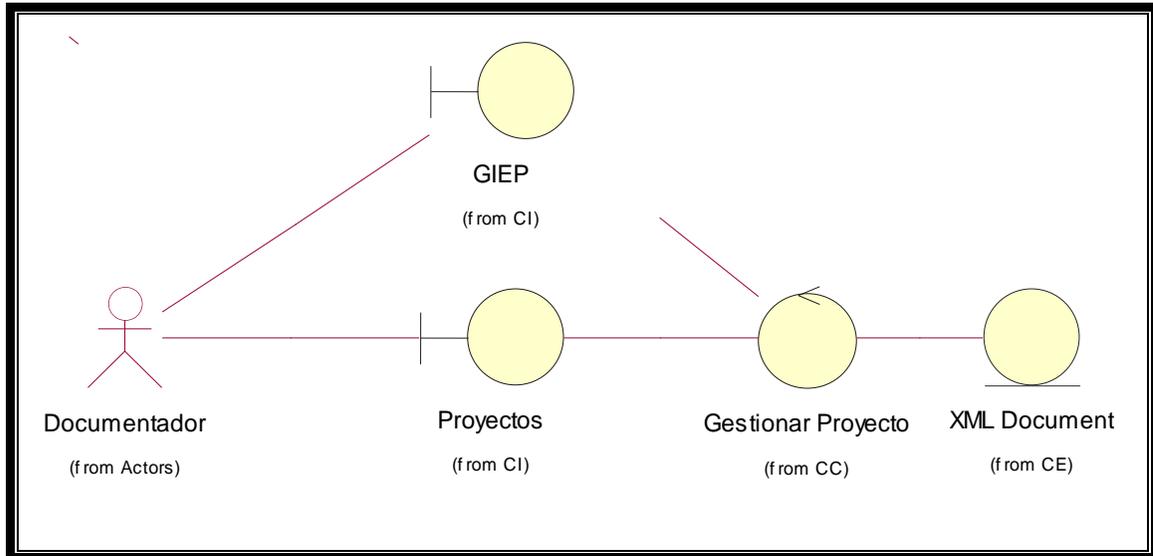


Diagrama 4 CUS Gestionar Proyectos

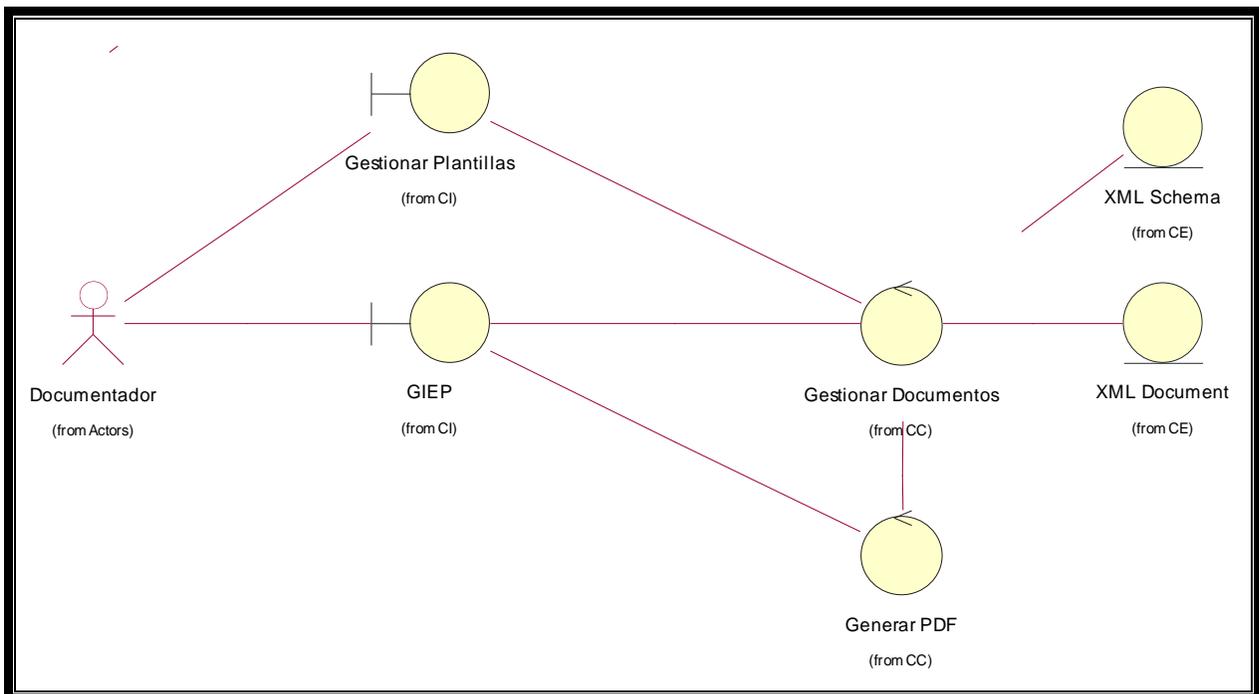


Diagrama 5 CU Gestionar Documentos

### **3.3. Modelo del Diseño**

El diseño dice como lograr los requerimientos relevados en el análisis, usando las metodologías definidas anteriormente, se puede describir, como el sistema llevará a cabo cada una de las funciones definidas, este es uno de las etapas más largas en el desarrollo de un sistema. La especificación del diseño aborda diferentes aspectos del modelo de diseño y se completa a medida que el diseñador refina su propia representación del software.

#### **3.3.1. Principios del Diseño**

##### **Interfaz de usuario**

- Se utilizará tipo de letra (Arial 11) para todos los textos que se agreguen o generen que no sea parte de la estructura de la plantilla PDF diseñada.
- La interfaz diseñada para la herramienta está basada en el estándar de ventanas.
- El tipo de letra a utilizar será Tahoma de estilo regular y tamaño 8.25 para el texto de los botones en la barra de herramientas y Microsoft Sans Serif de 8.25 para el resto de las etiquetas y mensajes de los formularios.
- El sistema debe mostrar una barra de herramientas para cumplir con las funcionalidades descritas para la misma.

##### **Ayuda**

La ayuda constará con toda la información referente al trabajo con las diferentes funcionalidades de la aplicación. Así como con una reseña de cómo estará estructurado el Expediente de Proyecto y su proceso de confección. Se podrá acceder a la misma por la barra de menú y con la opción de F1.

##### **Tratamiento de errores**

En el diseño de la interfaz se debe tener en cuenta el tratamiento de errores logrando que los mensajes de error que emita el sistema sean de fácil comprensión para el usuario y deben ser descriptibles para facilitar la corrección precisa y rápida.

### 3.4. Patrones

Conjunto de elementos que forman una unidad diferenciada y que se repiten a lo largo del tiempo, por lo que pueden tomarse como modelo o punto de referencia.

#### Fundamentación del uso de patrones.

Actualmente el uso de los patrones se ha generalizado, son utilizados y están presentes en casi todos los sistemas informáticos de importancia a nivel mundial. Los **patrones de diseño** (*design patterns*) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Los **GRASP** son patrones generales de software para asignación de responsabilidades, es el acrónimo de "General Responsibility Assignment Software Patterns". Aunque se considera que más que patrones propiamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software. (39)

**Experto:** Indica que la responsabilidad de la creación de un objeto siempre se debe asignar al experto en información, es decir, la responsabilidad recae sobre la clase que conoce toda la información necesaria para poder crearlo. El GRASP de experto en información es el principio básico de asignación de responsabilidades. Se indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo.

**Beneficios:** Se mantiene el encapsulamiento, los objetos utilizan su propia información para llevar a cabo sus tareas. Se distribuye el comportamiento entre las clases que contienen la información requerida. Son más fáciles de entender y mantener.

**Creador:** El patrón creador ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases.

La nueva instancia deberá ser creada por la clase que:

- Tiene la información necesaria para realizar la creación del objeto.
- Usa directamente las instancias creadas del objeto.
- Almacena o maneja varias instancias de la clase.

**Controlador:** El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado.

Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control.

Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

**Alta cohesión y bajo acoplamiento:** Se puede separar, aunque están íntimamente ligados, de hecho si se esfuerza en aumentar mucho la cohesión del software, es muy posible que se perjudique el acoplamiento aumentándolo, y por el contrario si se reduce mucho el acoplamiento, se verá disminuida la cohesión:

**Alta cohesión:** La información que almacena una clase debe de ser coherente y está en la mayor medida de lo posible relacionada con la clase.

**Bajo acoplamiento:** Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

### 3.4.1. Diagrama de clases del diseño

En el diagrama de clases de diseño se muestran los atributos y métodos de cada clase y se representa de una forma sencilla la colaboración y las responsabilidades de las distintas clases que forman el sistema.

Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema, esto incluye modelar el vocabulario del sistema, modelar las colaboraciones o modelar esquemas. Los diagramas de clases también son la base para un par de diagramas relacionados: los diagramas de componentes y los diagramas de despliegue. Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

### 3.5. Diagramas de clases del Diseño

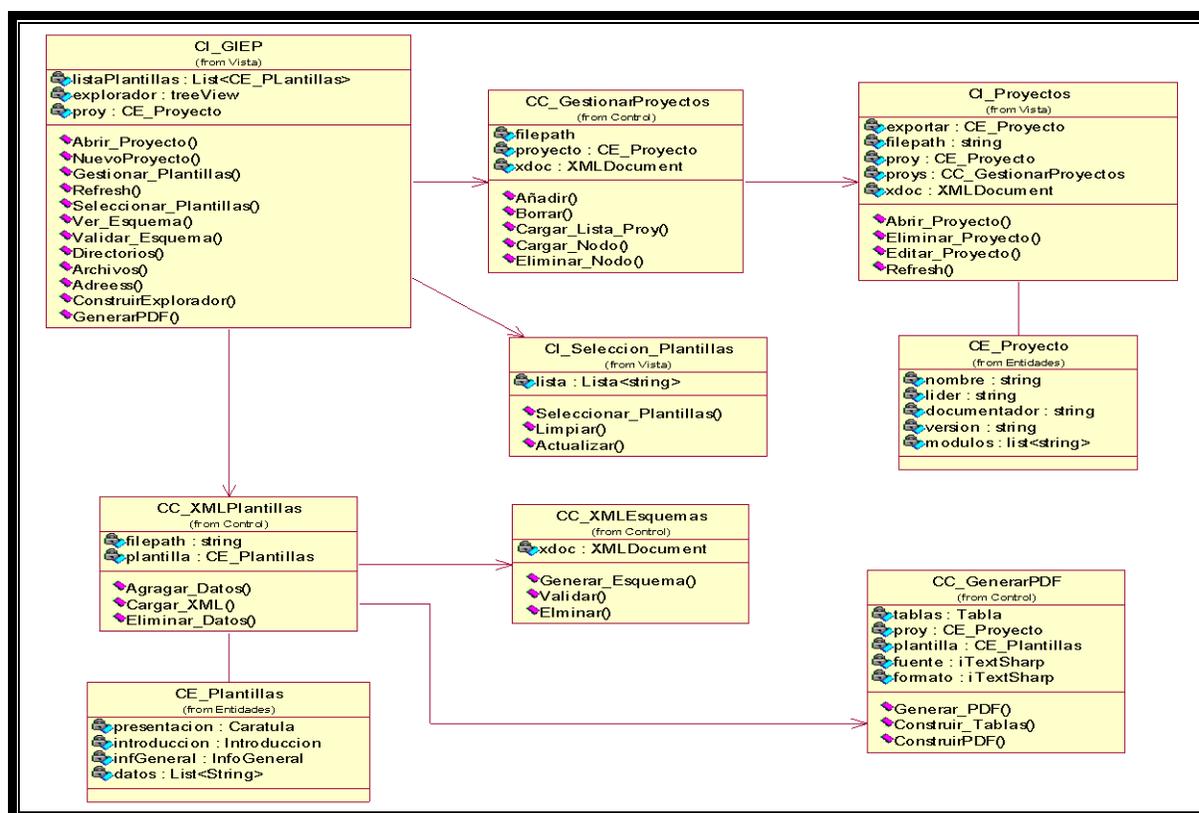


Diagrama 6 CU Gestionar Documentos

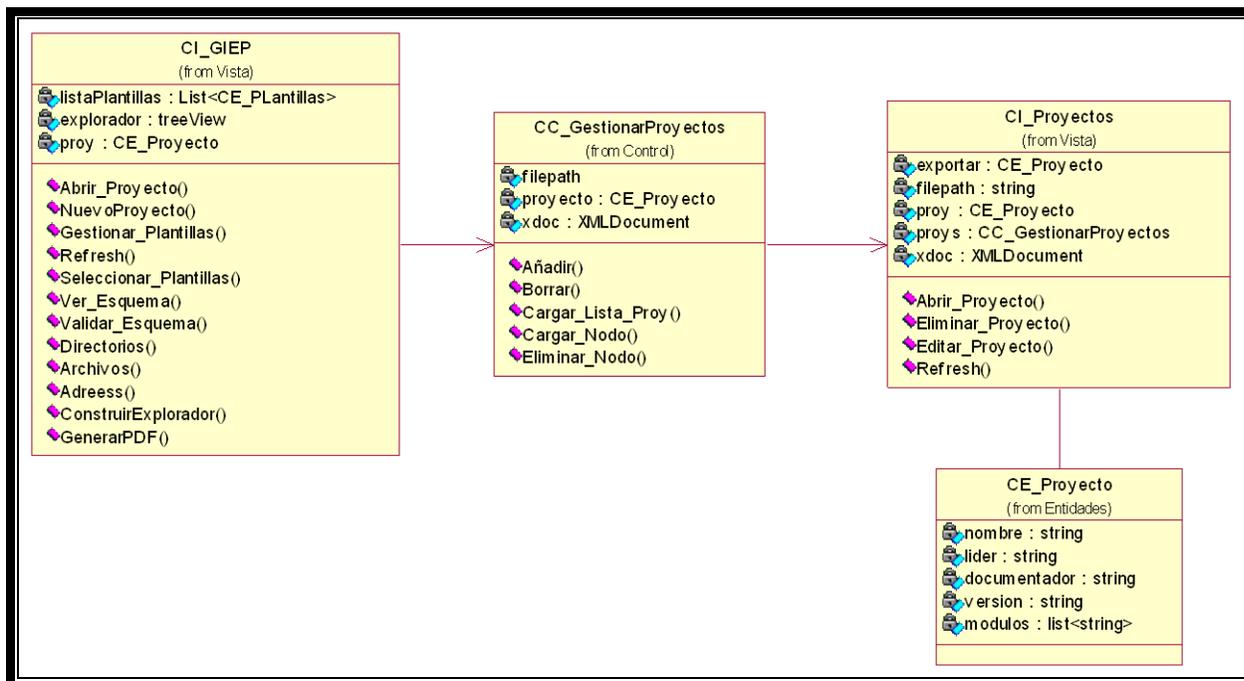


Diagrama 7 CU Gestionar Proyectos

### 3.6. Descripción de las clases

<b>Nombre:</b> CC_Gestionar_Proyectos	
<b>Tipo de clase (interfaz, controladora, entidad)</b>	Controladora
<b>Atributo</b>	<b>Tipo</b>
Líder_Proyecto	String
Documentador	String
Versión	String
Nombre_Proyecto	String
Módulos	String(String)
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	Agregar Módulos

<b>Descripción:</b>	Se agregan nuevos módulos de trabajo al proyecto en cuestión.
<b>Nombre:</b>	Eliminar Módulos
<b>Descripción:</b>	Se elimina un módulo o varios de la lista de módulos del proyecto en cuestión.
<b>Nombre:</b>	Modificar Líder
<b>Descripción:</b>	Se indicará el nombre del líder del proyecto.
<b>Nombre:</b>	Modificar Nombre del Proyecto
<b>Descripción:</b>	Se modificará el nombre del proyecto en caso de cambiar.

<b>Nombre:</b> CE_Proyecto_XML	
<b>Tipo de clase (interfaz, controladora, entidad--)</b>	Entidad
<b>Atributo</b>	<b>Tipo</b>
Líder Proyecto	String
Documentador	String
Versión	String
Nombre del proyecto	String
Módulos	String(String)
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	Crear XML_Proyecto
<b>Descripción:</b>	Se almacenará en un documento XML simple la información del EP que será redundante y la información general de las características del Proyecto en cuestión.

<b>Nombre:</b> CI_Proyecto	
<b>Tipo de clase (interfaz, controladora, entidad--)</b>	Interfaz
<b>Atributo</b>	<b>Tipo</b>

cproyecto	ComboBox
tblider	TextBox
tbdocum	TextBox
cmodulos	ComboBox
tbversion	TextBox
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	Agregar_Proyecto
<b>Descripción:</b>	Brindará la funcionalidad de mostrar el formulario para que el actor pueda insertar los nuevos proyectos con los cuales se trabajarán.
<b>Nombre:</b>	Agregar_MódulosXProyectos
<b>Descripción:</b>	Brindará la funcionalidad de mostrar el formulario para que el actor pueda insertar los nuevos módulos asociados a un proyecto con los cuales se trabajarán.
<b>Nombre:</b>	Listar_Proyectos
<b>Descripción:</b>	Brindará la funcionalidad de mostrar un listado de los proyectos existentes con sus respectivos módulos.

En el capítulo se detallaron las fases de Análisis y Diseño de la aplicación, donde se definen las clases y modelos que las conforman. Se especificaron los principios y patrones de diseño del sistema y se realizó una descripción de las clases del diseño de la aplicación. Además se detallan los tipos de datos o componentes que las conforman.

## 4. Capítulo IV. Implementación

Durante este capítulo se define como está organizado el sistema en términos de códigos y subsistemas organizados en capas. También se aborda la organización en términos de los elementos que componen la aplicación (ejecutables, códigos, archivos y otros).

### 4.1. Diagrama de Despliegue

Un Diagrama de Despliegue es un diagrama que muestra la configuración de los nodos que participan en la ejecución. Se modelan la topología del hardware sobre el que se ejecuta el sistema y la distribución física del sistema.

Para la puesta en práctica de la herramienta, solo se necesitan computadoras clientes donde correrá la aplicación en cada una. Las mismas utilizarán una impresora siendo esta un recurso compartido conectado a la red mediante el puerto USB.

### 4.2. Diagramas de componentes

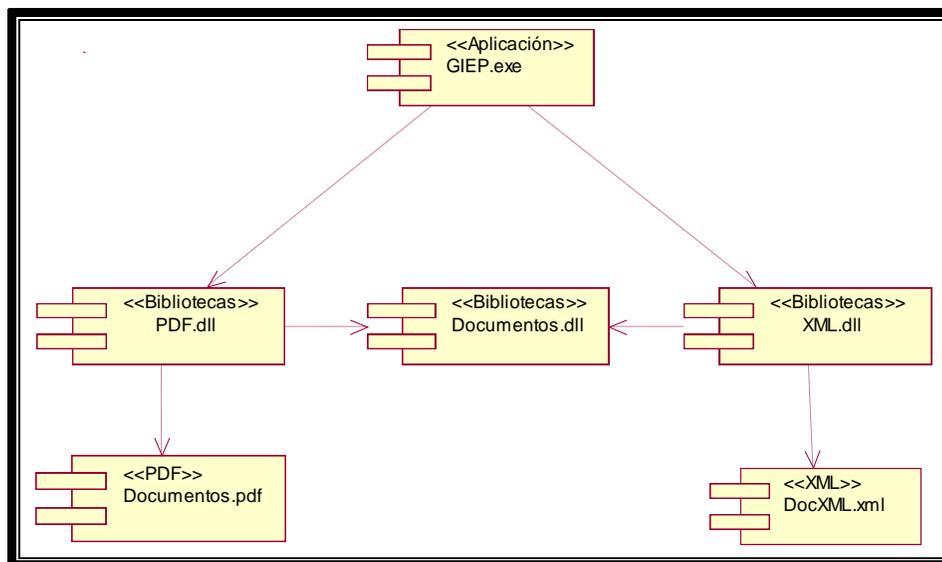


Diagrama 8 Diagrama de componentes

### 4.3. Implementación por Módulos

Con el objetivo de organizar el trabajo de implementación se estructuró la herramienta en pequeños módulos para el control de los diferentes artefactos con los que se relacionará la aplicación.

**Módulo para la gestión de los datos perdurables, los documentos XML:** Controlará el manejo de la información contenida en cada uno de los XML que corresponden a una plantilla determinada del EP. Una vez que en la aplicación se guarde la información de algún documento esta se salvará en su XML el cual guardará esta información organizada jerárquicamente (*Ver epígrafe: 4.3.1*). El módulo se encargará de Cargar y Guardar esta información. El componente para este módulo se llamará XML.dll.

**Módulo para el control de las clases Pozos de Datos:** Contendrá todas las plantillas del EP como almacén de datos. Los objetos para cada una de las plantillas serán usadas por los demás módulos o componentes de la herramienta, principalmente a través de los módulos de gestión de documentos XML y documentos PDF. El componente para este módulo se llama Documentos.dll.

**Módulo para la gestión de la información final, ficheros en formato PDF:** Se encargará de usar la información que retorna el módulo para la gestión de XML y transformará estos datos al formato de plantillas elaborado por el departamento de Calidad Central de la Universidad. El nombre de este componente es PDF.dll.

**Un módulo visual para interactuar con el usuario, el documentador:** Relacionará todos los componentes de la aplicación a través de la interfaz visual. El componente para este módulo será el componente principal de la aplicación, GIEP.exe.

#### 4.3.1. Descripción del Módulo de Gestión de datos perdurables (Documentos XML)

Los documentos XML estarán codificados bajo el estándar de codificación “**UTF-8**” en su **versión 1.0** de la World Wide Web Consortium (W3C):



Ilustración 1 Estándar de codificación de los XML según W3C

Y La información de todas las plantillas que conforman el Expediente de Proyecto UCI (EP) se guardará en documentos XML con una jerarquía organizada:

```

1 <Document_Root nombre = "Nombre de la Planilla">
2   <Informacion_Caratula>
3     <Polo nombre="Infraestructura Productiva" />
4     <Proyecto nombre="Nombre del proyecto" />
5     <Módulo nombre="Módulo al que pertenece la información" />
6     <Proyecto_Versión No.="Número de la versión del proyecto" />
7   </Informacion_Caratula>
8   <Resumen>
9     <Control_Versiones>
10      <Version No.="Número de la versión">
11        <Fecha>Fecha en la que se elabora la versión del documento</Fecha>
12        <Descripcion>Descripcion de la ficha de versión</Descripcion>
13        <Autor>El autor de la ficha</Autor>
14      </Version>
15    </Control_Versiones>
16    <Reglas_Confidencialidad>
17      <Clasificacion>Clasificación del documento</Clasificacion>
18      <Cliente nombre="Nombre del Cliente" />
19      <Propietario nombre="Nombre del Propietario" />
20    </Reglas_Confidencialidad>
21  </Resumen>
22  <Introducción>
23    <Proposito>Proposito del Documento</Proposito>
24    <Alcance>El alcance que tendrá el documento</Alcance>
25    <Abreviaturas>
26      <Abreviatura>listado de posibles abreviaturas que se utilizarán</Abreviatura>
27    </Abreviaturas>
28    <Referencias>
29      <Referencia>
30        <Codigo>Identificador de la Referencia</Codigo>
31        <Titulo>Titulo del documento, libro o artículo al que se hace referencia</Titulo>
32      </Referencia>
33    </Referencias>
34  </Introducción>
35  <Capítulo nombre = "Nombre del Capítulo">
36    <Información/> <!--Información del capítulo como tablas o datos simples-->
37  </Capítulo>
38 </Document_Root>

```

Ilustración 2 Estructura del Documento XML

#### 4.3.1.1. Implementación del Módulo XML

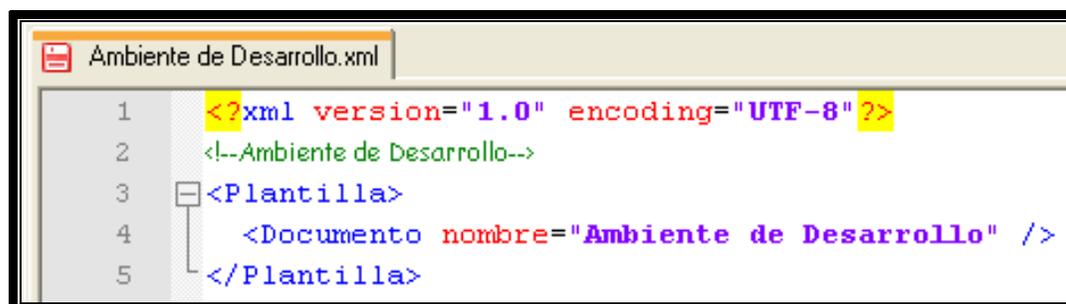
Para la creación de los documentos XML se utilizó la clase “Crear\_Plantillas\_XML.cs” que crea físicamente el documento con la extensión “.xml” y el nombre dinámicamente de la plantilla en cuestión; esta clase crea el ambiente del documento y establece la raíz del documento y el estándar de codificación para documentos XML:

```

20 public XmlDocument Plantilla_XML ()
21 {
22     CrearPath();
23     XmlDocument xdoc = new XmlDocument();
24     XmlTextWriter xmlWriter = new XmlTextWriter(fullpath,
25         System.Text.Encoding.UTF8);
26     xmlWriter.Formatting = Formatting.Indented;
27     xmlWriter.WriteProcessingInstruction("xml", "version='1.0' encoding='UTF-8'");
28     xmlWriter.WriteComment(planname);
29     xmlWriter.WriteStartElement("Plantilla");
30     xmlWriter.Close();
31     return xdoc;
32 }

```

Ilustración 3 Fragmento del código que crea el documento XML



```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--Ambiente de Desarrollo-->
3 <Plantilla>
4     <Documento nombre="Ambiente de Desarrollo" />
5 </Plantilla>

```

Ilustración 4 Resultado del código para crear documentos XML

Para cada uno de las plantillas fue creada una clase para la lectura y escritura de los documentos XML, estas clases cumplen con los objetos correspondientes en el módulo de pozos de datos; en ellas existen dos métodos importantes Agregar\_Datos y Cargar\_Datos; el primero imprimirá en los documentos con la

estructura de código XML los datos que le sean pasados y el segundo leerá el documento XML y guardará los datos contenidos en sus nodos en el nuevo objeto correspondiente al documento.

```

public void Agregar_Datos()
{
    XmlElement nodo1 = xdoc.CreateElement("Nombre_del_Nodo1");
    XmlText textonodo1 = xdoc.CreateTextNode("Datos_para_nodo1");
    nodo1.AppendChild(textonodo1);

    XmlElement nodo2 = xdoc.CreateElement("Nombre_del_Nodo2");
    nodo2.SetAttribute("nombre", "Datos_del_Nodo2");
}
public object Cargar_Datos()
{
    object doc = new object();
    doc.nodo1 =
        xdoc.GetElementsByTagName("Nombre_del_Nodo1")[0].InnerText;
    doc.nodo2 =
        xdoc.GetElementsByTagName("Nombre_del_Nodo2")[0].
            GetAttribute("nombre");
}

```

Ilustración 5 Ejemplo de Métodos para la lectura y escritura de nodos en un documento XML

#### 4.3.1.2. Estándar de codificación UTF-8 de la W3C

**UTF-8** (8-bit Unicode Transformation Format) es una norma de transmisión de longitud variable para codificación de caracteres | caracteres codificados utilizando Unicode, creada por Rob Pike y Ken Thompson. **UTF-8** usa grupos de bytes para representar el estándar de Unicode para los alfabetos de muchos de los lenguajes del mundo. Es especialmente útil para la transmisión sobre sistemas de correo de 8 bits.

Usa de 1 a 4 bytes por carácter, dependiendo del símbolo de Unicode. Por ejemplo, se necesita un solo byte en UTF-8 para codificar los 128 caracteres ASCII|US-ASCII en el rango U+0000 a U+007F de Unicode.

Aunque podría parecer ineficiente representar caracteres Unicode con hasta 4 bytes, UTF-8 permite a sistemas antiguos transmitir caracteres de este súper conjunto de ASCII. Además, sigue siendo posible utilizar compresión de datos independientemente del uso de UTF-8. (40)

### *Ventajas:*

- Por supuesto, la ventaja más notable de cualquier Formato de Transformación Unicode sobre codificaciones heredadas es que este puede codificar cualquier carácter.
- Algunos símbolos Unicode (incluido el Alfabeto latino) se tomarán como 1 byte, aunque otros pueden tomar más de 4. Así, UTF-8 generalmente ahorrará espacio comparado con UTF-16 o UTF-32 en donde los caracteres ASCII de 7 bits son comunes.
- El primer byte de una secuencia multi-byte es suficiente para determinar la longitud de una secuencia multi-byte-. Esto hace extremadamente simple extraer una sub-cadena de una cadena dada sin elaborar un análisis exhaustivo.
- La mayor parte del software existente (incluido el sistema operativo) no fueron escritos con Unicode en mente, y usar Unicode con ellos podría crear algunos problemas de compatibilidad. Por ejemplo, la librería estándar del lenguaje de programación C marca el final de una cadena con el carácter de un único byte 0x00. En la codificación Unicode UTF-16 la letra A en inglés es codificada como 0x0041. La librería considerará el primer byte 0x00 como el fin de la cadena e ignorará el resto. UTF-8, sin embargo, está diseñado para que los bytes codificados nunca tomen alguno de los valores de los caracteres especiales de ASCII, previniendo estos problemas y otros similares.
- Las cadenas en UTF-8 pueden ser ordenadas usando rutinas de ordenamiento estándar orientadas a byte (sin embargo no habrá ninguna diferenciación entre las mayúsculas y las minúsculas con representaciones que exceden el valor 128).
- UTF-8 es el valor predeterminado para el formato XML.

### **4.3.2. Descripción del Módulo de Gestión de pozo de datos**

Este módulo controla los objetos con los que van a interactuar las demás clases. Cada plantilla tendrá un objeto al que le hará referencia y toda la información que se guarda en los XML sale de los datos que se acumulen en estos objetos, dichos datos los escribirá en estos objetos el mismo componente para el trabajo con documentos XML y el componente para la impresión de los documentos en formato PDF.

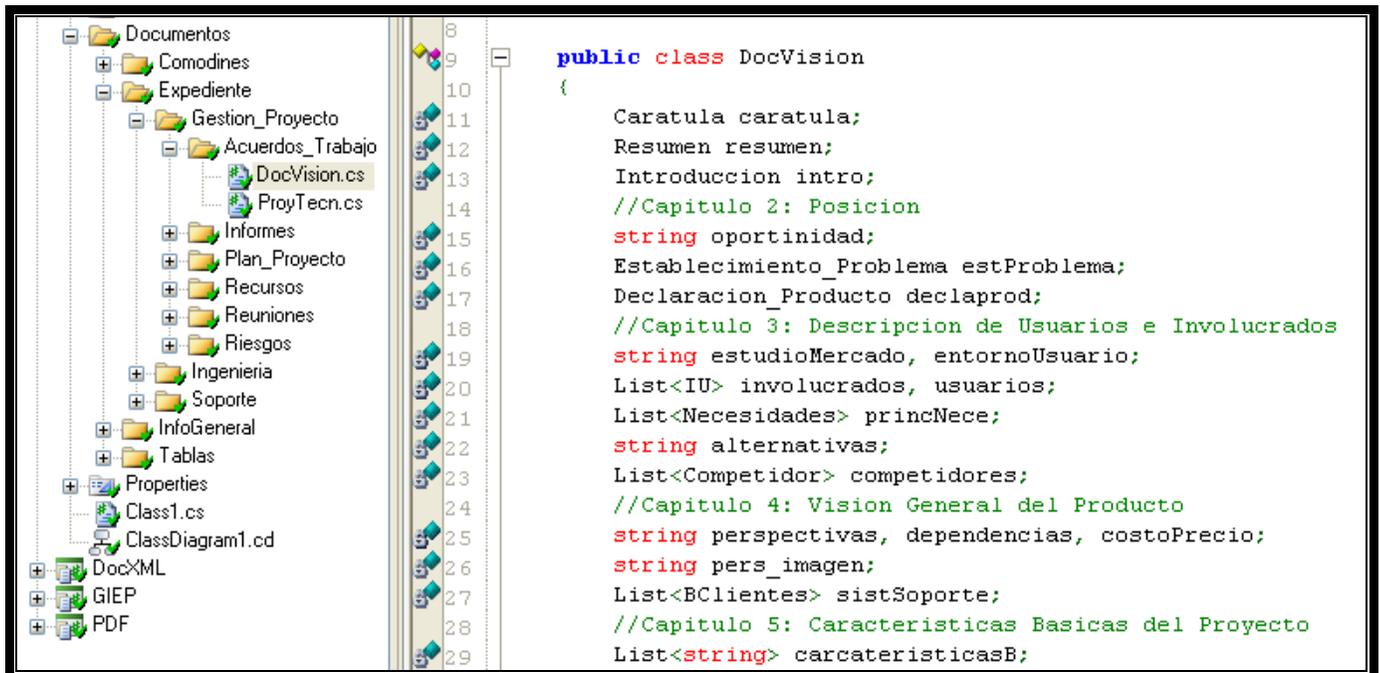


Ilustración 6 Objeto para la representación del Documento Visión

Las plantillas de los documentos no solo contendrán datos simples si no que ellos a la vez poseerán listados de otros tipos de objetos, estos últimos serán objetos para indicar un elemento de una tabla. Para estas se agruparon una serie de objetos para la descripción del contenido de las tablas.

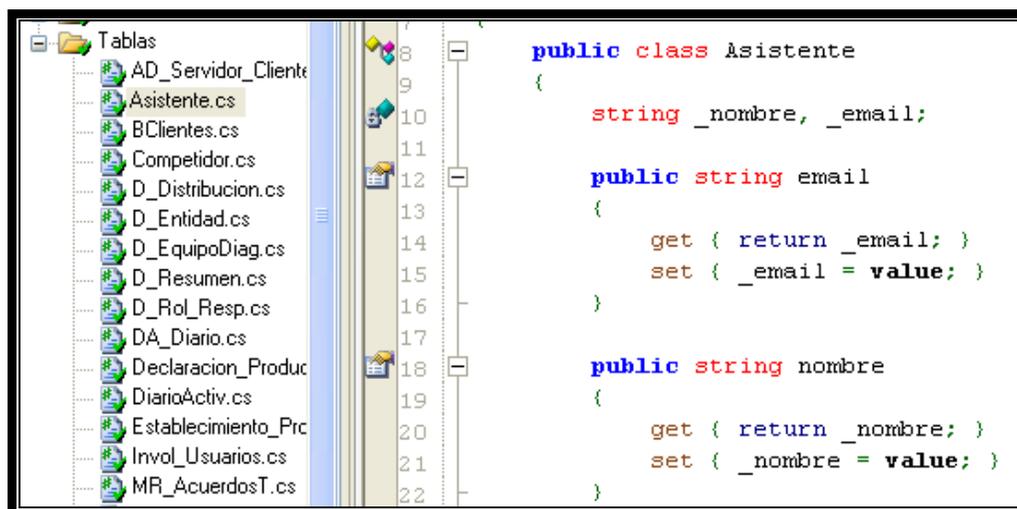


Ilustración 7 Objeto para la tabla de participantes a una reunión del documento *Minutas Reunión*

### 4.3.3. Descripción del Módulo Visual (GIEP.exe)

La aplicación visualiza en un panel lateral los documentos pertenecientes a cada módulo de un proyecto en específico, para dicha visualización se usó el componente visual *treeView* para lograr una vista lógica de la estructura del EP.

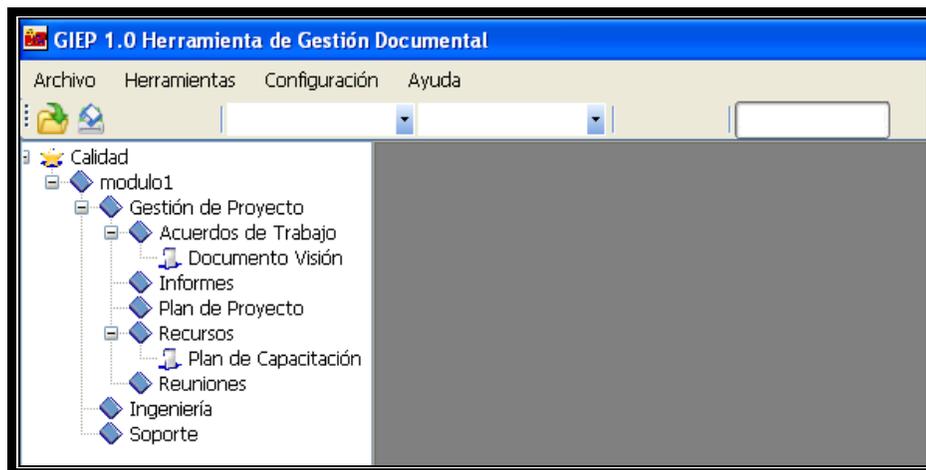


Ilustración 8 El panel lateral izquierdo muestra la jerarquía de carpetas del EP con el uso de un *treeView*

Para la zona de trabajo se hizo uso de diferentes *UsserControl* que se reutilizarían en los diferentes documentos especificándose cada uno en el momento de su visualización.

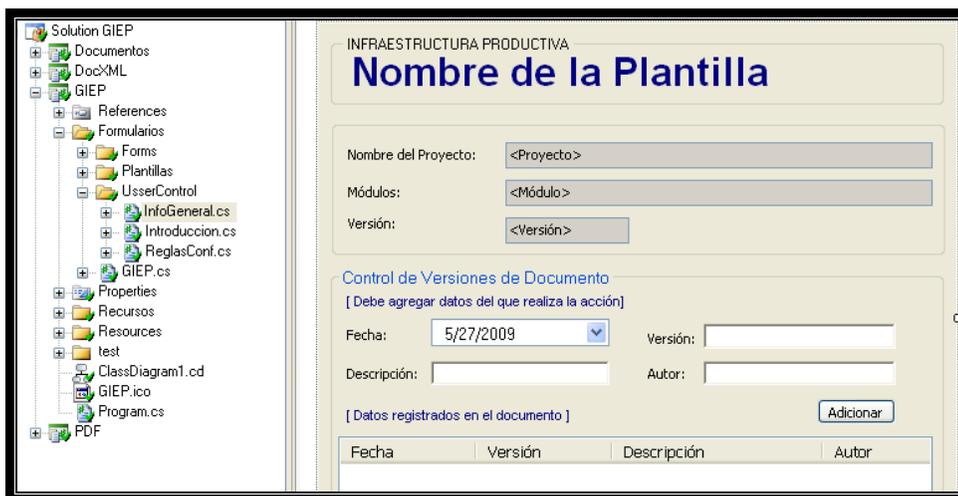


Ilustración 9 Vista en el SharpDevelop del *UsserControl* "InfoGeneral" para mostrar la Información General

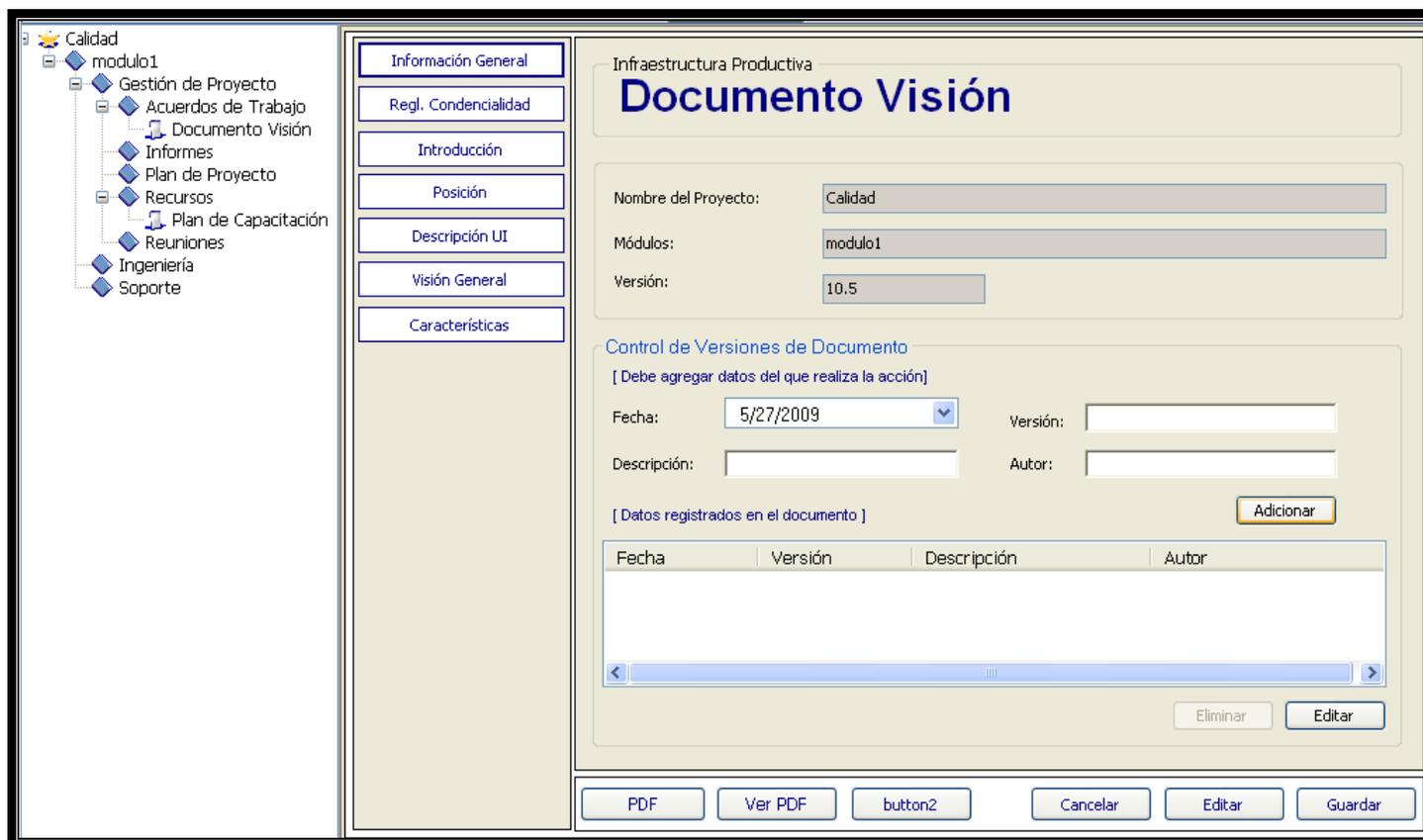


Ilustración 10 Información General del Documento Visión en tiempo de ejecución de la aplicación

#### 4.3.3.1. Sub-Módulo de PrintScreen

Por la necesidad de la toma de pantallazos (PrintScreen) para incluir anexos a los documentos se desarrolló una herramienta interna en la aplicación que captura pantallazos con un tamaño de imagen personalizado, permitiendo al usuario no tener que editar pantallazos tomados por teclado con otro software. Se le da la comodidad al usuario de no tener que salir de la aplicación para la captura de imágenes para los anexos. Al llamar a esta herramienta la aplicación principal se hace transparente para no estorbar en la captura de los pantallazos.

El código para esta herramienta fue tomada de la contribución de otros programadores que postean resultado de sus conocimientos en sitios públicos para el intercambio del conocimiento informático, uno de estos sitios es [www.codeproject.com](http://www.codeproject.com) del cual salió la idea de esta herramienta.

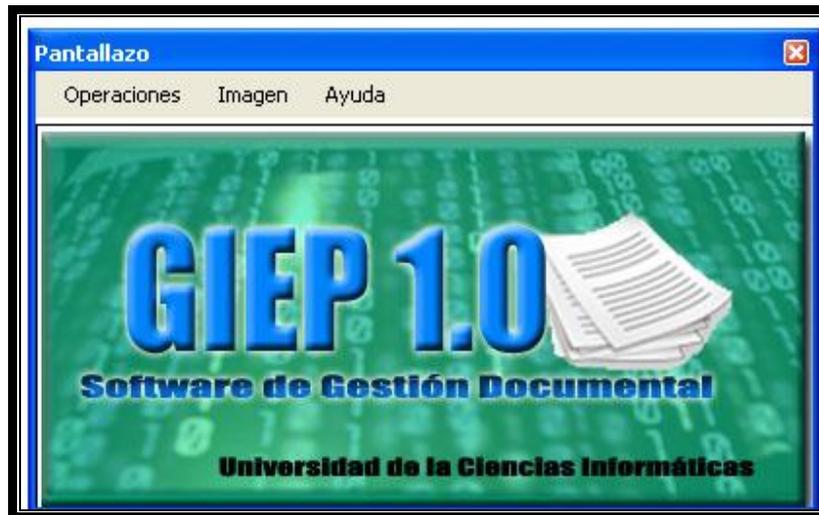


Ilustración 11 Vista de la herramienta interna para la toma de pantallazos (PrintScreen)

#### 4.3.3.2. Sub-Módulo de Ayuda

Con el objetivo de ayudar al usuario en el trabajo con la aplicación se desarrolló un pequeño módulo interno en Robo-Help, sirviendo de guía para la realización de las diferentes acciones que se pudiesen realizar..

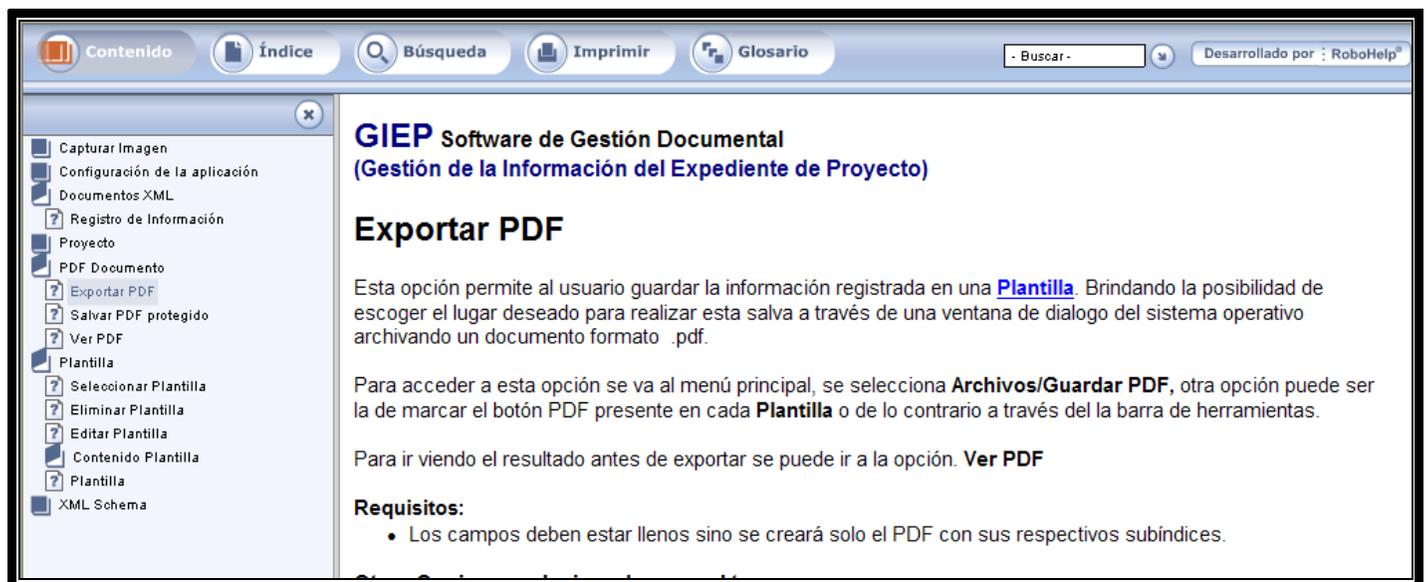


Ilustración 12 Vista del asistente de Ayuda

#### 4.3.4. Descripción del Módulo para generar documentos en formato PDF

Este módulo se encarga de convertir toda la información contenida en un documento XML en un documento con formato PDF basado en las plantillas definidas por el departamento de Calidad Central de la UCI.

Se hizo uso de la librería *iTextSharp.dll* como referencia para el diseño de las plantillas que conforman el EP, esta brinda facilidades para el trabajo con la información y su exportación a diferentes formatos y dentro de ellos PDF.

##### 4.3.4.1. Librería *iTextSharp*

*iTextSharp.dll* es una librería que permite generar archivos PDF sobre la marcha; es ideal para los desarrolladores que buscan mejorar la web y otras aplicaciones con la dinámica de generación de PDF y / o su manipulación. iText no es una herramienta final para los usuario sino por el contrario, se puede utilizar iText para construir otras aplicaciones con el objetivo de automatizar la creación de PDF y la manipulación del proceso. Por ejemplo, en uno o más de las siguientes situaciones:

- Que debido a restricciones de tiempo o de tamaño, los documentos PDF no pueden ser producidos manualmente.
- El contenido del documento debe ser calculado, o debe estar basado en las entradas de información de los usuarios.
- El contenido necesita ser adaptado o personalizado.

Esta librería se puede usar para:

- Visualizar los documentos PDF desde los navegadores web.
- Generar documentos dinámicamente desde archivos XML o bases de datos.
- Agregar a los documentos PDF los números de página, marcas de agua, imágenes, etc.
- Separar, concatenar y/o manipular páginas PDF o los documentos en sí.
- Automatización del llenado de formularios en el PDF.
- Añadir firmas digitales a un archivo PDF.

En resumen: las clases iText son muy útiles para personas que necesitan generar documentos de sólo lectura, independiente de la plataforma de documentos que contengan texto, listas, tablas e imágenes o que quieren realizar determinadas manipulaciones en los documentos PDF. La biblioteca es especialmente útil en combinación con Java, basados en la tecnología de Servlets; también hay una. NET puerto disponible: iTextSharp (escrito en C #).

iTextSharp esta soportado bajo la licencia LGPL<sup>28</sup> y MPL<sup>29</sup>.

#### 4.3.4.2. Creación de documentos PDF en GIEP

Para la creación de los documentos PDF se obtiene toda la información correspondiente a una plantilla determinada, en este se escribe con la función *PdfWriter* todo el contenido de dicha plantilla.

```
{  
    PdfWriter.GetInstance(document, new FileStream(filepath, FileMode.OpenOrCreate));  
}
```

Ilustración 13 Línea de código que crea el documento PDF

Se diseñó una clase para cada una de las plantillas del EP, ellas contienen la información para el diseño de dichas plantillas detallando la información por capítulos como tablas, imágenes, entre otros campos que no son comunes para todas las plantillas. El contenido común se define en una clase, *PDFRecurrente.cs* la cual es usada en las plantillas que lo requieran. El formato para el diseño de las tablas y otros campos, es el establecido por el departamento de Calidad Central de la UCI: tamaño y tipo de letra, colores de los campos y textos, alineación y espaciado, entre otros.

---

<sup>28</sup> **Licencia Pública General Menor** de GNU, o más conocida por su nombre en inglés **GNU Lesser General Public License**, es una licencia de software creada por la Free Software Foundation. Los contratos de licencia de la mayor parte del software están diseñados para jugar con su libertad de compartir y modificar dicho software. En contraste, la "*GNU General Public License*" pretende garantizar la libertad de compartir y modificar el software "libre", esto es para asegurar que el software sea libre para todos sus usuarios.

<sup>29</sup> **Mozilla Public License** es una licencia de código abierto y software libre. Fue desarrollada originalmente por Netscape Communications Corporation (una división de la compañía América Online) y más tarde su control fue traspasado a la Fundación Mozilla.

```

public Phrase Portada(Caratula c)
{
    Phrase pag_uno = new Phrase();
    Paragraph p = new Paragraph();
    pag_uno.Add(new Chunk("\n\n\n\n\n\n" + "INFRAESTRUCTURA PRODUCTIVA"+
        "\n", Fuente.NegritaB_10));
    pag_uno.Add(new Chunk("\n\n" + c.Docname.ToString() + "\n\n\n".ToString(),
        Fuente.Azul_18));
    pag_uno.Add(new Chunk(c.ProyName.ToString() + "\n", Fuente.AzulB));
    pag_uno.Add(new Chunk(c.ProyMod.ToString() + "\n", Fuente.AzulB));
    pag_uno.Add(new Chunk(c.ProyVersion.ToString(), Fuente.AzulB));
    p.Add(pag_uno);
    p.Alignment = 2;
    return p;
}

```

Ilustración 14 Método para el diseño de la portada de las plantillas

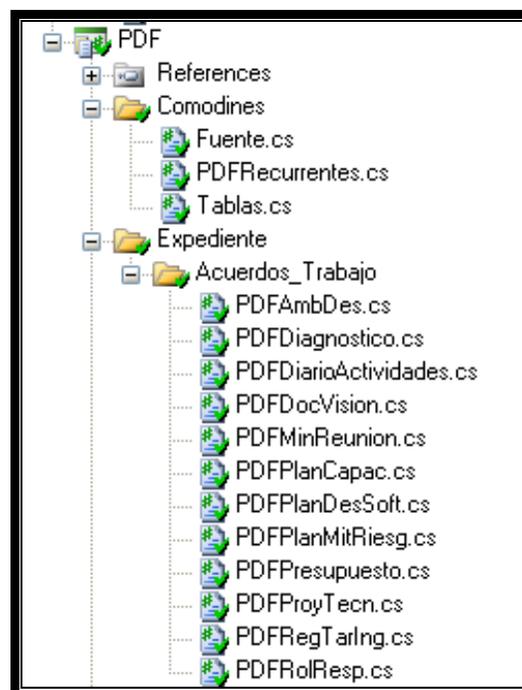


Ilustración 15 Vista del explorador de las clases que diseñan las plantillas

### 4.4. Herramienta para la Gestión de la Información del Expediente de Proyecto (GIEP)

Con el uso de las librerías que pertenecen al framework .net 2.2 de la Microsoft y otras librerías como iTextSharp se desarrolló una herramienta para la automatización del contenido de las plantillas del EP, la cual da solución a la problemática planteada en este trabajo y facilita la elaboración del EP con una mejor calidad y eficiencia. La herramienta es de fácil uso y posee un tiempo de respuesta de las acciones acorde con los requerimientos del hardware especificados. La estructuración de la aplicación en pequeños módulos permite que se puedan realizar cambios en el código fuente sin afectar el resto de los componentes o en caso de afectación, esta sería mínima; facilitando la inclusión de mejoras a la aplicación para futuras iteraciones.

#### Ventajas que brinda GIEP

- Elimina el exceso de información recurrente.
- Ahorra tiempo en la confección de los documentos.
- Facilita la revisión de los documentos para el equipo de desarrollo y el equipo de calidad.
- Formato compatible con la mayoría de los sistemas operativos (PDF).
- Permite la captura y gestión de imágenes.
- Cálculos automatizados.
- Rápido acceso a la información del EP.

En el Capítulo se describió la estructura y organización del sistema mediante un diagrama de Componentes y se detalló la estructura de la aplicación del sistema en módulos, ofreciéndose características específicas de las librerías de clases utilizadas y su aplicación para alcanzar las funcionalidades deseadas.

## **Conclusiones**

Las soluciones informáticas existentes para la Gestión Documental de la información en los proyectos de desarrollo de software no responden a las necesidades del proceso de elaboración del Expediente de Proyecto de la Universidad de las Ciencias Informáticas.

Actualmente el proceso de elaboración del Expediente de Proyecto, se realiza en los editores de textos (Microsoft Word, Open Office) presentando problemas a la hora de su entrega. Se hace necesaria la implementación de una aplicación que de solución a todos estos problemas.

Se seleccionaron las tecnologías, metodologías y herramientas más adecuadas para la implementación de la herramienta.

Se definió una arquitectura de diseño e implementación para la elaboración de la aplicación.

Se obtuvo como resultado final una herramienta que contribuye a resolver las dificultades en la elaboración del Expediente de Proyecto y llenado de su información.

## Recomendaciones

Los autores recomiendan:

Incluir un estudio para la realización de las pruebas requeridas al software.

Incorporar un módulo para el tratamiento de errores ortográficos y sintácticos en el contenido de las plantillas.

Implementación de nuevos módulos que permitan obtener los documentos en otros formatos como DOC, ODT, XSL y otros compatibles con los sistemas a los que se lleve la aplicación.

Integrar la aplicación con un sistema de control de versiones como el Subversion u otras aplicaciones que permitan la gestión documental basada en versiones.

Crear una alternativa web para el trabajo online entre diferentes proyectos con el objetivo de optimizar el tiempo de entrega de los Expedientes de Proyectos a los clientes y grupos de revisión.

## Referencias Bibliográficas

1. Gestión Documental. [En línea] 2005.  
[http://www.guellconsulting.com/servicios/internet/gestion\\_documental](http://www.guellconsulting.com/servicios/internet/gestion_documental). Tesis “Estructura del contenido de las plantillas del Expediente de Proyecto”
2. *Monográfico Calidad del Software / Software de calidad*. Número 137, s.l.: NOVATICA, Enero-Febrero 2004.
3. Gestión Documental. [En línea] 2005.  
[http://www.guellconsulting.com/servicios/internet/gestion\\_documental](http://www.guellconsulting.com/servicios/internet/gestion_documental).
4. *Monográfico Calidad del Software / Software de calidad*. Número 137, s.l.: NOVATICA, Enero-Febrero 2004.
5. Adapting Document [En línea] 2009. <http://www.adapting.com/aplicaciones/gestiondocumental/>
6. ADOC LTDA [En línea] 2009 <http://www.adoc.com.co/pages/quienes.htm>
7. El sitio web de ZyLAB [En línea] 2009  
[http://www.zylab.es/indexes/index\\_ZyIMAGE\\_Information\\_Access\\_Platform.htm](http://www.zylab.es/indexes/index_ZyIMAGE_Information_Access_Platform.htm)
8. [En línea] <http://www.catalogodesoftware.com/producto.aspx?pid=451>
9. Siam IRS [En línea] 2009 <http://www.siamirs.es>
10. 102NOVADOC. Empresa [En línea] 2009 <http://www.102novadoc.es/>
11. Getsa. Ingeniería Documental [En línea] 2009 <http://www.gedsa.es/software-gestion.php>.
12. Eíto Brun, Ricardo. Documentum lanza Edms'98. [En línea] 2000. <http://www.documentum.com>
13. García Caballero, Ricardo. *Gestión de los documentos electrónicos en los nuevos servicios de información y documentación*. [ed.] Bonifacio Martín Galán. Universidad Carlos III de Madrid: Everest.

14. Ver referencia 13
15. Ver referencia 13
16. Ver referencia 13
17. Pressman, R.S. Ingeniería del Software: Un enfoque práctico. 2002.
18. González Hernández, Yanisbel y Sánchez Caballero, Lyna Isabel; *Estructura del contenido de las plantillas del Expediente de Proyecto*; [UCI] 2008
19. *Project Description*. [En línea] 2006. <http://plone.org/products/plone> .
20. *New Summer Mambo templates released*. [En línea] 2004. <http://www.mamboserver.com>.
21. *Drupal 5.1 and 4.7.6 released*. [En línea] 2007. <http://drupal.org>.
22. *Welcome to the OpenCms Project*. [En línea] 2006. <http://www.opencms.org/opencms/en>.
23. *Trabajo documental y colaborativo*. [En línea] 2006.  
[http://www.csi.map.es/csi/tecniap/tecniap\\_2006/02T\\_PDF/trabajo%20documental.pdf](http://www.csi.map.es/csi/tecniap/tecniap_2006/02T_PDF/trabajo%20documental.pdf).
24. [http://www.gitdoc.com/ctl\\_arch/razones.htm](http://www.gitdoc.com/ctl_arch/razones.htm) [En línea] 2009
25. *DME -Document Management Extension*. [En línea] 2007.  
[http://www.bs.com.ar/bsweb/MENU/SUBMENU/Archivos/bs\\_053.pdf](http://www.bs.com.ar/bsweb/MENU/SUBMENU/Archivos/bs_053.pdf).
26. Collins- Sussman, B. *Control de versiones con Subversion*. 2004. pág. 286. 9. *Los sistemas de gestión integral de la documentación en las organizaciones. Métodos de Información*. Allepuz Ros, Teresa y Gutierrez la Rubia, Carme. 5, mayo 2001, Vol. 2, págs. 29-33.
27. Ver referencia 19
28. Ver referencia 19
29. Mitchell, W. D. (2001). *Java sin errores*. Madrid : McGraw-Hill: ISBN: 84-481-3107-1

30. Microsoft Corporation. MSDN Microsoft Developer Network. *Visual Studio Developer Center*. [En línea] Microsoft Corporation. [Citado: 5 21, 2008.] <http://msdn.microsoft.com/en-gb/vstudio/bb265237.aspx>
31. Archer, Tom. *A FONDO C#*. Redmond : McGraw-HILL/INTERAMERICANA DE ESPAÑA, S.A.U, 2001. ISBN: 84-481-3246-7
32. Ver referencia 30
33. Ver referencia 30
34. Borland Software Corporation. C# Builder. *Borland C# Builder Developer Resources*. [En línea] Borland Software Corporation. [Citado: 5 21, 2008.] <http://www.csharpbuilder.info>
35. ic#code. ic#code. *SharpDevelop The Open Source Development Environment for .NET*. [En línea] ic#code. [Citado: 5 21, 2008.] <http://www.icsharpcode.net/OpenSource/SD/>
36. MonoDevelop. MonoDevelop Free GNOME Development Environment. *Feature List*. [En línea] MediaWiki. [Citado: 5 21, 2008.] [http://www.monodevelop.com/Feature\\_List](http://www.monodevelop.com/Feature_List)
37. 2007. [En Línea] <https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducci%C3%B3n%20a%20RUP.doc>
38. Rational Enterprise Edition, Ayuda extendida. 2003
39. *Larman, Craig*. UML y Patrones. 2007. Capítulo 16 (GRASP: Diseño de objetos con responsabilidades)
40. **UTF-8 and Unicode Standards** <http://www.utf-8.com/>

## Bibliografía

- 102NOVADOC. (2009). Obtenido de <http://www.102novadoc.es/>
- *Adapting Document*. (2009). Obtenido de <http://www.adapting.com/aplicaciones/gestiondocumental/>
- ADOC LTDA. (2009). Obtenido de <http://www.adoc.com.co/pages/quienes.htm>
- Allepuz Ros, T. y. (2001). *Los sistemas de gestión integral de la documentación en las organizaciones. Métodos de Información*.
- Archer, T. (2001). *A FONDO C#*. McGraw-HILL/INTERAMERICANA DE ESPAÑA.
- Borland Software Corporation. (2008). *C# Builder. Borland C# Builder Developer Resources*. Obtenido de <http://www.csharpbuilder.info>
- *Catálogo de Software*. (s.f.). Obtenido de <http://www.catalogodesoftware.com/producto.aspx?pid=451>
- Collins- Sussman, B. (2004). *Control de versiones con Subversion*.
- *DME -Document Management Extension*. (2007). Obtenido de [http://www.bs.com.ar/bsweb/MENU/SUBMENU/Archivos/bs\\_053.pdf](http://www.bs.com.ar/bsweb/MENU/SUBMENU/Archivos/bs_053.pdf)
- *Drupal 5.1 and 4.7.6 released*. (2007). Obtenido de <http://drupal.org>.
- Eíto Brun, R. (2000). *Documentum lanza Edms'98*. Obtenido de <http://www.documentum.com>
- García Caballero, R. *Gestión de los documentos electrónicos en los nuevos servicios de información y documentación*. Universidad Carlos III de Madrid: Everest: Bonifacio Martín Galán.
- *Gestión Documental*. (2005). Obtenido de Gestión Documental
- *Getsa. Ingeniería Documental*. (2009). Obtenido de <http://www.gedsa.es/software-gestion.php>
- *GITDOC*. (2009). Obtenido de [http://www.gitdoc.com/ctl\\_arch/razones.htm](http://www.gitdoc.com/ctl_arch/razones.htm)
- González Hernández, Y. y. (2008). Estructura del contenido de las plantillas del Expediente de Proyecto. UCI.
- ic#code. (2008). *SharpDevelop The Open Source Development Environment for .NET*. Obtenido de <http://www.icsharpcode.net/OpenSource/SD/>

- *Introducción a RUP*. (s.f.). Obtenido de <https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducci%C3%B3n%20a%20RUP.doc>
- Larman, C. (2007). *UML y Patrones*.
- Microsoft Corporation. (2008). *MSDN Microsoft Developer Network. Visual Studio Developer Center*. Obtenido de <http://msdn.microsoft.com/en-gb/vstudio/bb265237.aspx>
- Mitchell, W. D. (2001). *Java sin errores*.
- *MonoDevelop Free GNOME Development Environment*. (2008). Obtenido de [http://www.monodevelop.com/Feature\\_List](http://www.monodevelop.com/Feature_List)
- Monográfico Calidad del Software / Software de calidad. (Enero-Febrero de 2004). *Número 137, s.l.: NOVATICA* .
- *New Summer Mambo templates released*. (2004). Obtenido de <http://www.mamboserver.com>.
- Pressman, R. (2002). *Ingeniería del Software: Un enfoque práctico* .
- *Project Description*. (2006). Obtenido de <http://plone.org/products/plone>
- Rational Enterprise Edition, Ayuda extendida. (2003).
- *Siam IRS* . (2009). Obtenido de <http://www.siamirs.es>
- *Trabajo documental y colaborativo*. (2006). Obtenido de [http://www.csi.map.es/csi/tecniap/tecniap\\_2006/02T\\_PDF/trabajo%20documental.pdf](http://www.csi.map.es/csi/tecniap/tecniap_2006/02T_PDF/trabajo%20documental.pdf)
- *UTF-8 and Unicode Standards* . (s.f.). Obtenido de <http://www.utf-8.com/>
- *Welcome to the OpenCms Project*. (2006). Obtenido de <http://www.opencms.org/opencms/en>.
- *ZyLAB "Sitio Oficial"*. (2009). Obtenido de [http://www.zylab.es/indexes/index\\_ZyIMAGE\\_Information\\_Access\\_Platform.htm](http://www.zylab.es/indexes/index_ZyIMAGE_Information_Access_Platform.htm)