

Universidad de las Ciencias Informáticas

Facultad 7



Componente para la edición del formato de impresión de los documentos generados en la variante Web del sistema de Gestión de la Información del Expediente de Proyecto

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

Autores: Adel Fernández Garrido

Damián Almeida García

Tutores: Ing. Anet Conde Almeida

Ing. Dariel Fernando Reyes Prieto

La Habana, 30 de Junio de 2011

“Año 53 de la Revolución”

*“Toda ciencia empieza en la
imaginación
y no hay sabio sin el arte de
imaginar.”*

José Martí

DATOS DE CONTACTO

Tutores:

Tutor: Ing. Anet Conde Almeida

Cotutor: Ing. Dariel Fernando Reyes Prieto

Tutor principal:

Ing. Anet Conde Almeida:

Graduada en Ingeniería en Ciencias Informáticas en el año 2007 en la UCI. Categoría Docente: Instructor. Ha impartido las asignaturas de Matemática I, Matemática II, Ingeniería de Software II y Ética Informática. Pertenece al grupo de Mercadotecnia del CESIM. aconde@uci.cu

Co tutor

Ing. Dariel F. Reyes Prieto:

Profesor graduado de Ingeniero en Ciencias Informática en el año 2009 en la UCI. Categoría docente: Instructor Recién Graduado. Ha impartido las asignaturas Ingeniería de Software. Se desempeña como jefe de Pruebas en el grupo de Calidad del CESIM. dfreyes@uci.cu

Autores:

Adel Fernández Garrido

Universidad de las Ciencias Informáticas, Habana, Cuba

Email: agarrido@estudiantes.uci.cu

Damián Almeida García

Universidad de las Ciencias Informáticas, Habana, Cuba

Email: dagarcia@estudiantes.uci.cu

RESUMEN

En el Centro de Informática Médica existen varios proyectos de desarrollo de software para los Sistemas de Salud Pública de Cuba y otros países latinoamericanos. La gestión de la información del expediente de proyecto se realiza en documentos de texto plano. En estos documentos, la información se repite o son concurrentes por lo que la gestión de los cambios se hace engorrosa o es susceptible de errores. Otro de los problemas es la aplicación de formato a los documentos.

Todo esto trae consigo la necesidad de crear un sistema informatizado que permita la gestión de la información del expediente de proyecto. La estructura del expediente de proyecto cambia constantemente, por lo que es necesario agregar en el sistema un componente para la edición. El objetivo del presente trabajo es desarrollar un componente que permita editar el formato de impresión de los documentos generados en el sistema Gestión de Información del Expediente de Proyecto en la variante Web.

La aplicación se desarrolla sobre tecnologías Microsoft Visual Studio 2008, C# como lenguaje de programación, Framework .NET de Microsoft, Enterprise Architect como herramienta CASE y RUP¹ como metodología de desarrollo. Finalmente se obtuvo un componente que le permite al usuario crear o modificar plantillas del expediente de proyecto así como permitirle al usuario darle formato de impresión y mostrándole una vista previa.

Palabras claves:

Componente, documentos, esquemas XML, formato, impresión.

¹Proceso Unificado Racional (Rational Unified Process).

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1. CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA	5
1.2. APLICACIONES WEB, VENTAJAS Y DESVENTAJAS.....	6
1.3. HERRAMIENTAS WEB PARA LA EDICIÓN DEL FORMATO DE IMPRESIÓN	8
1.4. HERRAMIENTAS Y TECNOLOGÍAS.....	13
1.5. METODOLOGÍA DE DESARROLLO DE SOFTWARE	21
1.6. FORMATOS DE LOS DOCUMENTOS A INCORPORAR EN LA APLICACIÓN	23
1.7. FORMATO DE IMPRESIÓN	25
1.8. TECNOLOGÍAS Y HERRAMIENTAS A UTILIZAR	26
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA	27
2.1. OBJETO DE ESTUDIO.....	27
2.2. OBJETO DE AUTOMATIZACIÓN	28
2.3. PROPUESTA DEL SISTEMA.....	28
2.4. DESCRIPCIÓN DEL MODELO DEL DOMINIO.....	28
2.5. ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE	30
2.6. REQUERIMIENTOS FUNCIONALES DEL SISTEMA.....	30
2.7. REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA.....	32
2.8. DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA.....	34
2.9. DEFINICIÓN DE LOS ACTORES.....	35
2.10. DESCRIPCIÓN EXPANDIDA DE LOS CASOS DE USO.....	36
CAPÍTULO 3. DISEÑO DEL SISTEMA	42
3.1. ARQUITECTURA	42
3.2. DISEÑO	43
3.3. DIAGRAMA DE CLASES DEL DISEÑO.....	45
3.4. DESCRIPCIÓN DE LAS CLASES DEL DISEÑO	49
CAPÍTULO 4. IMPLEMENTACIÓN	50

4.1.	MODELO DE IMPLEMENTACIÓN	50
4.2.	DIAGRAMA DE DESPLIEGUE	50
4.3.	COMPONENTES	51
4.4.	DIAGRAMA DE COMPONENTES	51
4.5.	TRATAMIENTO DE ERRORES	52
CONCLUSIONES		54
RECOMENDACIONES.....		55
REFERENCIAS BIBLIOGRÁFICAS		56
BIBLIOGRAFÍA		59
ANEXOS.....		62
	ANEXO 1: DIAGRAMAS DE CLASES DEL DISEÑO	62
	ANEXO 2:DIAGRAMAS DE INTERACCIÓN (SECUENCIA).....	65
	ANEXO 3:PANTALLAS DEL SISTEMA	68
GLOSARIO DE TÉRMINOS.....		73

INTRODUCCIÓN

La evolución de la Informática y las Telecomunicaciones en las últimas dos décadas ha colocado a los sistemas informáticos en un rol preponderante dentro de las instituciones, siendo un fenómeno tecnológico importante que ha dotado a la humanidad de medios potentísimos para la expansión del conocimiento, la cultura y la sabiduría. Esta situación se ha visto apoyada por el creciente auge de Internet, subdesarrollo y la habitual dependencia de los productos informáticos cada día más actualizados, potentes y con mayor calidad.

Las empresas que se dedican a la producción de software con grandes cantidades de información, a veces les resulta difícil alcanzar una adecuada gestión y calidad de la misma, afectando de forma general su producción. Para mantener y garantizar el manejo adecuado de estas informaciones surge el término Gestión Documental, que no es más que el uso de tecnologías y procedimientos que permitan la gestión y el acceso unificado a la información generada en la organización. La Gestión Documental se encarga de dar una solución para el manejo cuidadoso de los documentos que se deseen preservar en las empresas.

Actualmente la industria del software a nivel mundial cada vez es más competente y culta. Es por ello, que Cuba no está exenta dentro de esta gran gama de desarrollo científico y tecnológico. En función de su inserción en el mercado internacional, ha destinado cuantiosos recursos, esfuerzos y tecnologías para agilizar y potenciar el desarrollo de las investigaciones en el campo de la Informática. De esta forma la industria del software nacional tiene la tarea de liberar productos de una elevada calidad y eficiencia para poder colocarse en los primeros peldaños a nivel mundial en el campo de la Informática.

En diferentes sectores políticos, económicos y sociales, el país ha obtenido buenos resultados. Poco a poco, se han introducido en las diferentes empresas el uso de las computadoras, con fines económicos y en aras de un perfeccionamiento profesional.

Uno de los grandes proyectos de Cuba, es la creación de la Universidad de las Ciencias Informáticas (UCI), institución en la que se logra el principio de estudio-producción a través de la vinculación a proyectos productivos de los estudiantes. Está constituida por siete facultades docentes en la sede central y 3 facultades regionales, vinculadas a un área específica de la economía en Cuba. Tiene dentro de sus principales objetivos impulsar la informatización de la sociedad e ir a la vanguardia en el desarrollo de

software. Para lograr estos lineamientos posee una estrategia centrada en la creación de los grupos de Calidad, regidos por la Dirección de Calidad de Software. Estos están orientados al logro de la excelencia en las soluciones informáticas de los proyectos productivos existentes en cada facultad.

El grupo de Calidad de la Facultad 7 está orientado a agilizar los procesos de pruebas que en él se realizan. Lo que está determinado por la existencia de múltiples proyectos que abarcan diferentes departamentos. En estos departamentos, a la par del desarrollo de los productos de software, deben generar un conjunto de documentos durante todo el transcurso del proceso de desarrollo, los cuales son agrupados en el Expediente de Proyecto (EP). Estos documentos son sometidos a revisión, pues constituyen un elemento de gran importancia en el ciclo de vida que da lugar al producto final. (1)

En la actualidad en la Facultad 7 existe una herramienta para la Gestión de Información del Expediente de Proyecto llamada (GIEP) en su versión 1.0. La misma provee varios servicios y opciones para gestionar la información generada por los proyectos de la facultad, debido a que es una aplicación de escritorio se plantea la necesidad de realizar una versión web de la herramienta GIEP que agilice la elaboración del EP. Permitiendo la interacción de los usuarios mediante la red y la integración a las nuevas herramientas de Gestión de Proyecto, también la necesidad de agregarle nuevas funcionalidades que faciliten el trabajo y su uso práctico.

La anterior versión de GIEP presenta opciones que permiten la exportación de las plantillas a formato pdf, pero la configuración de su estructura se realizaba a código, por lo que surge la necesidad de agregarles nuevas funcionalidades que permitan la estructuración de estas plantillas de forma visual. Y además crear configuraciones personalizadas por proyectos de acuerdo a las necesidades particulares de cada uno. También es necesario agregarles nuevos formatos a exportar y que sean compatibles a los diferentes sistemas operativos existentes en la universidad, permitiendo una mayor operabilidad y aplicación.

Dada la situación anterior, el **problema a resolver** consiste en: ¿Cómo desarrollar un componente para que la variante Web del sistema de Gestión de la Información del Expediente de Proyecto GIEP permita editar el formato de impresión de los documentos?

Como **objeto de estudio** se identifica, los procesadores de textos y el **campo de acción** se enmarca en la edición del formato de impresión de los documentos en los procesadores de texto en la Web.

Para dar solución al problema antes mencionado se propone como **objetivo general**: Desarrollar un componente que permita editar el formato de impresión de los documentos generados en el sistema Gestión de Información del Expediente de Proyecto en la variante Web.

Para cumplir con el objetivo anteriormente se proponen las siguientes **tareas de la investigación**:

- ✓ Evaluar las tecnologías y librerías existentes que puedan ser usadas en el trabajo.
- ✓ Analizar los formatos de los documentos en los sistemas operativos utilizados en la Universidad.
- ✓ Seleccionar metodología de desarrollo, lenguaje de programación y la herramienta de modelado a utilizar.
- ✓ Definir la prioridad de los componentes a implementar en el sistema.
- ✓ Elaborar la documentación correspondiente a los Flujos de Trabajo propuestos por la metodología seleccionada.
- ✓ Implementar las funcionalidades requeridas en la aplicación.

El presente trabajo cuenta de cuatro capítulos estructurados de la siguiente forma:

Capítulo 1: Fundamentación Teórica

Se aborda el estado del arte del tema a tratar a nivel internacional, nacional y de la universidad e incluye una explicación de las técnicas, tecnologías, metodologías y software empleados en la investigación para darle solución al problema. Se tratan los conceptos fundamentales para una correcta comprensión del tema.

Capítulo 2: Características del sistema

Se define las características del sistema y se enmarca en el objeto de estudio. Se hace un análisis del dominio de la aplicación, se describen los procesos para darle solución al problema y se generan documentos referentes a esta fase. También se definen requisitos no funcionales.

Capítulo 3: Diseño del sistema

Primeramente se realiza el diseño del sistema donde se define el diagrama de clases del diseño por casos de uso así como la relación existente entre ellas. Posterior a esto se muestra la interacción entre los actores

y el sistema mediante los diagramas de secuencia. Además se especifica la seguridad y el diseño de la interfaz de la aplicación.

Capítulo 4: Implementación.

Se presenta el modelo de implementación del Componente para la edición del formato de los documentos generados en la variante WEB del sistema de Gestión de la Información del Expediente de Proyecto.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En este capítulo se aborda el tema de los procesadores de textos, edición del formato de impresión en la web y las diferentes herramientas existentes a nivel nacional e internacional que permiten realizar este servicio. Se realiza un análisis de estas herramientas para poder identificar las diferentes ventajas y desventajas de cada uno de ellos y también el porqué de la necesidad de crear una herramienta web para los diferentes tipos de reportes y la edición del formato de impresión, así como la justificación de las metodologías y tecnologías a usar para la elaboración de la misma.

1.1. Conceptos asociados al dominio del problema

Una **aplicación** es un programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo. Suele resultar una solución informática para la automatización de ciertas tareas complicadas como pueden ser la contabilidad o la gestión de un almacén. Ciertas aplicaciones desarrolladas 'a medida' suelen ofrecer una gran potencia ya que están exclusivamente diseñadas para resolver un problema específico.(2)

Un **procesador de texto** es una aplicación informática destinada a la creación o modificación de documentos escritos por medios de una computadora. Representa una alternativa moderna a la antigua máquina de escribir, siendo mucho más potente y versátil. Los procesadores de texto brindan unas u otras posibilidades según la aplicación de que se disponga.

Como regla general básica, todos los procesadores de texto pueden trabajar con distintos formatos de párrafo, tamaño y orientación de las fuentes, efectos de formato, además de contar con las propiedades de poder cortar y copiar texto, fijar espacio entre líneas y entre párrafos, alinear párrafos, establecer sangrías y tabulados, crear y modificar estilos, activar presentaciones preliminares antes de la impresión o visualizar las páginas editadas. Los procesadores de texto incorporan desde hace algunos años también correctores automáticos de ortografía y gramática así como diccionarios multilingües y de sinónimos que facilitan sobremanera la labor de redacción.(3)

Las **Plantillas** son una forma de dispositivo que proporciona una separación entre la forma o estructura y el contenido. Es un instrumento que permite guiar, portar o construir un diseño o esquema predefinido.

Mediante la manipulación de los **formatos de impresión** se pueden cambiar los documentos que genera el sistema de Gestión de la Información del Expediente de Proyecto, para que cumplan con las normas exigidas por la ley y para hacerlos parecidos a los formatos que se manejan en las empresas.

Un **esquema XML** no es más que 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.(4)

1.2. Aplicaciones WEB, ventajas y desventajas

Con la llegada de la Web 2.0², propiciada esta por el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) y los nuevos retos de Internet, se abre paso una nueva idea en la concepción del software; SaaS³ es un modelo de distribución del software que proporciona a los clientes el acceso al mismo a través de la red (generalmente Internet), de manera que les libra del mantenimiento de las aplicaciones, de operaciones técnicas y de soporte.

Las aplicaciones distribuidas en la modalidad SaaS pueden llegar a cualquier tipo de empresa sin importar su tamaño o su ubicación geográfica. Se trata de un modelo que une el producto (software) al servicio, para dotar a las empresas de una solución completa que permita optimizar sus costes y sus recursos. (CIIN) (5)

Ventajas de la WEB:

- ✓ **Compatibilidad multiplataforma:** Las aplicaciones Web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables. Varias tecnologías incluyendo Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales.

²El término **Web 2.0** fue acuñado por Tim O'Reilly en 2004 para referirse a una segunda generación en la historia de la Web basada en comunidades de usuarios y una gama especial de servicios, como las redes sociales, los blogs y los wikis, que fomentan la colaboración y el intercambio ágil de información entre los usuarios.

³ Software orientado a servicios (SaaS acorde a sus siglas en inglés).

- ✓ Actualización: Las aplicaciones basadas en Web están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo, pues no se hace necesario iniciar nuevas descargas y/o procedimientos de instalación (algunas veces imposible cuando el usuario está trabajando dentro de grandes organizaciones).
- ✓ Inmediatez de acceso: Las aplicaciones basadas en Web no necesitan ser descargadas, instaladas y configuradas. Ellos acceden a su cuenta online y están listas para trabajar sin importar cuál es su configuración o su hardware. (5)
- ✓ Menos requerimientos de memoria: Las aplicaciones basadas en Web tienen menos demandas de memoria RAM de parte del usuario final que los programas instalados localmente. Al residir y correr en los servidores del proveedor, esas aplicaciones basadas en Web usan en muchos casos la memoria de las computadoras donde ellas corren, dejando más espacio para correr múltiples aplicaciones sin incurrir en frustrantes deterioros en el rendimiento.
- ✓ Precio: Las aplicaciones basadas en Web no requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software descargable tradicional. Esto permite que las aplicaciones online cuesten una fracción de sus contrapartes descargables si no totalmente gratuitas, mientras que ofrecen componentes adicionales y servicios Premium como una opción.
- ✓ Los datos también van online: El hecho de que el manejo de los datos sea realizado de forma remota libra al usuario de la responsabilidad en la protección de los mismos, y al mismo tiempo logra que los recursos sean accesibles en cualquier momento.
- ✓ Múltiples usuarios concurrentes: Las aplicaciones basadas en Web pueden ser utilizadas por múltiples usuarios al mismo tiempo. No hay más necesidad de compartir pantallas o enviar instantáneas cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta. (6)

Desventajas de la WEB:

- ✓ Acceso limitado, la necesidad de conexión permanente y rápida a Internet hacen que el acceso a estas aplicaciones no esté al alcance de todos.

- ✓ La interactividad no se produce en tiempo real, en las aplicaciones Web cada acción del usuario conlleva un tiempo de espera hasta que se obtiene la reacción del sistema.
- ✓ Elementos de interacción muy limitados. En comparación con el software de escritorio, las posibilidades de interacción con el usuario que ofrecen las aplicaciones Web (mediante formularios principalmente) son muy escasas.

Diferencias de presentación entre plataformas y navegadores. La falta de estándares ampliamente soportados dificulta el desarrollo de las aplicaciones.(7)

1.3. Herramientas WEB para la edición del formato de impresión

1.3.1. *Ámbito internacional*

CollabEdit:

Es una herramienta que permite escribir un documento de texto cualquiera al mismo tiempo que otros usuarios pueden ver dichos cambios en tiempo real y editar dichos cambios. Además la herramienta permite aplicarle formato al texto para resaltar palabras clave dependiendo del lenguaje de programación en el que esté escrito.

Lo que destaca a esta herramienta del resto es su simplicidad y facilidad de uso. No requiere registrarse ni pagar por el uso del servicio además que no define permisos de vista o edición, todos se vuelven editores del documento. Al entrar al sitio aparece a primera vista un botón para crear un nuevo documento el cual puede ser compartido con otras personas copiando la URL⁴ única que se genera para cada documento.

La desventaja de su simplicidad es que:

- ✓ Al no tener la necesidad de registrarse en el sitio para poseer una cuenta, tampoco se posee la funcionalidad de guardar el documento aunque sí permite descargar el contenido como archivo.

⁴Localizador de Recurso Uniforme (Uniform Resource Locator).

- ✓ El propósito más apropiado para esta herramienta es entonces, el uso como medio para discutir código fuente que ya se tiene creado mediante algún IDE⁵ de desarrollo pero se necesita discutirlo o corregirlo de manera colaborativa con otros colegas a través de internet.
- ✓ El usarlo como medio para discutir código fuente que ya se tiene creado mediante algún IDE de desarrollo pero se necesita discutirlo o corregirlo de manera colaborativa con otros colegas a través de internet.

Cacoo:

Permite crear diagramas y ser editados de forma colaborativa con varios usuarios a la vez. A diferencia de Collabedit, este si requiere que se registren en su sitio pero a cambio se goza de las ventajas de poder asignar permisos a la hora de compartir tus diagramas además de que estos quedan guardados en su cuenta.

Posee figuras para la creación de una gran variedad de diagramas, como por ejemplo UML⁶, Diagramas de Flujo, Diagramas de Red, Sitemaps, etc. Posee herramientas que lo asemejan a una aplicación de escritorio como el copy/paste de figuras, y exportación a PNG⁷. Además de todo lo anterior, el workspace cuenta con una sección para chat en la que los editores del diagrama pueden discutir como si estuvieran en el Google Talk, sin la necesidad de moverse a otra ventana.

Google Docs y Hojas de cálculo:

Oficialmente es un programa gratuito basado en Web para crear documentos en línea con la posibilidad de colaborar en grupo. Incluye un procesador de textos, una hoja de cálculo, programa de presentación básico y un editor de formularios destinados a encuestas.

Características:

⁵Entorno integrado de desarrollo (Integrated Development Environment).

⁶Lenguaje Unificado de Modelado (Unified Modeling Language).

⁷ Formato de compresión de imágenes.

- ✓ Se pueden crear documentos de texto, dibujos, hojas de cálculo, presentaciones y bases de datos desde la misma aplicación o importarlos utilizando su interfaz web o enviándolos por el correo electrónico, además de subir cualquier tipo de archivo.
- ✓ Los archivos se almacenan en los servidores de Google. Los archivos almacenados pueden ser exportados en diversos formatos estándar o ser enviados por correo electrónico.
- ✓ Durante la edición de los documentos, éstos se guardan automáticamente para evitar pérdida de información.
- ✓ Otra característica importante es la posibilidad de colaboración de grupos de trabajo, además de poder compartirlos con múltiples usuarios al mismo tiempo.
- ✓ Google Docs es actualmente soportado por Google Chrome, Mozilla Firefox, Internet Explorer, Opera y Safari, entre otros.

Limitaciones:

- ✓ Existen ciertos límites al manejar documentos por cada cuenta. Los documentos de texto pueden tener hasta 500 KB más 2 MB para imágenes incrustadas. Cada hoja de cálculo puede tener hasta 10.000 filas, 256 columnas, 100.000 celdas y 40 hojas. Sólo pueden abrirse hasta 11 hojas al mismo tiempo e importarse presentaciones de hasta 10 MB.

YourDraft:

Es un editor de textos online el cual no requiere de java ni de flash porque es un editor de textos completamente WYSIWYG⁸, con las suficientes herramientas necesarias como si se estuviera en un procesador de textos local.

Con el funcionamiento similar a Flexlist (crea bases de datos simples de lo que se quiera, con todos los campos que se necesiten), no se necesita registrarse sino más bien poner el título, el correo electrónico de manera opcional y poniendo el código, donde se puede generar el texto que se desee. Para ello se tiene

⁸Lo que ves es lo que obtienes (What You See Is What You Get).

una dirección que debe de ser secreta, con la que se acceder, además se tienen dos direcciones más, una con la que compartir los documentos para su sola visualización y otra para el mismo fin que además pueda editar el texto.

Lo más importante es que se pueden insertar documentos flash en el documento de texto y que la función de los botones está en español a pesar de ser un servicio en inglés.

Buzzword:

Es un procesador de texto semejante a Word u Open Office, pero accesible desde el navegador web, con una elegante y amigable interfaz, ideal para que un equipo que no cuente con las herramientas necesarias pueda resolver una situación dada.(8)

Es necesario registrarse dentro del sitio para poder utilizarlo, pero realmente éste es un paso algo tedioso, puesto que una vez que se llena el formulario de registro, inmediatamente se podrá utilizar dicho servicio sin necesidad de abrir el correo para activar la cuenta de usuario.

Writely:

Es un procesador de textos, como si a Word o a AbiWord lo convierte en una aplicación web con acceso mediante el navegador, un interfaz Ajax y varias características adicionales como son la posibilidad de añadir etiquetas (tags) a los documentos y la de compartir (ya sea en lectura o también en escritura) con otros usuarios.

Writely sigue el camino de otras aplicaciones como GMail o Flickr, demostrando como una aplicación web puede resultar tan completa como una aplicación de escritorio y añadir características valiosas como son las del acceso a los documentos desde cualquier lugar y desde distintas plataformas.

Cyberprint:

Es un software de Serpol Software (una reconocida empresa que diseña software para empresas) que permite obtener un archivo con una bitácora de la actividad de una o más impresoras, así como monitorear y avisar por email cuando exista algún problema con las impresiones. Este software se instala en cada PC que tenga una impresora conectada directamente (Local) o en Red y para tener acceso a la configuración

de Cyberprinter se requiere de una contraseña con el fin de evitar que cualquier persona pueda hacer cambios a esta configuración.

Entre sus características generales se puede destacar que registra datos como nombre de documento, número de páginas impresas, copias, nombre de la impresora, fecha, hora, nombre de la PC, nombre de usuario, color de la impresión, tamaño de la hoja. Es el único contador de impresiones protegido con contraseña para evitar accesos no autorizados a la configuración del mismo. Solicita contraseña para desactivar el programa. Puede enviar la bitácora de impresión diariamente a cualquier buzón de email. De la misma forma puede enviar la bitácora de impresión diariamente a un servidor de páginas Web.

Realiza reportes estadísticos de impresión por día o rango de días y reportes de uso por usuario, por PC y por impresora. Puede asociar costos de impresión dependiendo del color seleccionado (Color o Monocromático). Igualmente asocia costos de impresión por tamaño de la hoja, nombre del documento o nombre de la impresora. Exporta la bitácora de impresión a su hoja de cálculo favorita. Funciona con impresoras láser, inyección de tinta y matriz y con impresoras paralelas, impresoras USB, servidores de impresión, impresoras con su propia tarjeta de red, impresoras inalámbricas e impresoras remotas en Internet.

1.3.2. Ámbito nacional

En la Universidad hoy en día, al igual que el resto de las instituciones que desarrollan software en el país, se realizan diferentes acciones para incidir en la mejora del proceso de gestión de la documentación en el expediente de proyecto. Para ello se comienzan a utilizar algunas herramientas de las antes mencionadas. A su vez se han llevado a cabo algunas investigaciones dirigidas fundamentalmente a buscar otras soluciones que se adapten a las características particulares de los procesos que se llevan a cabo en la Universidad.

1.3.3. Análisis crítico de las herramientas

Una vez realizado el estudio de las herramientas expuestas se llega a la conclusión de que muchas de ellas son propietarias y los costes de soporte técnico y mantenimiento son muy elevados, además algunas son de licencia gratis y código libre, por lo que no responden a las necesidades que se requiere para dicho

problema. Para darle solución al mismo es necesario desarrollar una herramienta que sea capaz de suplir dichas necesidades, siendo factible realizar un estudio de la tecnología y herramientas para su elaboración.

1.4. Herramientas y Tecnologías

1.4.1. Lenguaje 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 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, porque es diferente programar para red local, para Windows, Mac o Linux. A continuación se presentarán algunas características, ventajas y desventajas del lenguaje de programación seleccionado para trabajar.

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. 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-334⁹ de la Ecma International¹⁰.(9)

C# es un lenguaje orientado a objetos, con seguridad de tipos que permite a los desarrolladores crear una amplia gama de aplicaciones sólidas y seguras que se ejecutan en .NET Framework¹¹. Puede utilizar este lenguaje para crear aplicaciones cliente para Windows tradicionales, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, y otras tareas.(9)

⁹Estándar internacional que especifica la representación y semántica de programas escritos en C# así como la sintaxis y restricciones de este lenguaje.

¹⁰Organización internacional que se basa en membresías de estándares para la comunicación y la información.

¹¹El Microsoft .NET Framework, es un componente de software que puede ser o es incluido en los sistemas operativos.

C# admite los conceptos de encapsulación, herencia y polimorfismo. Una clase puede heredar directamente de una clase primaria, pero puede implementar cualquier número de interfaces. En C#, una estructura es como una clase sencilla; es un tipo asignado en la pila que puede implementar interfaces pero que no admite la herencia.

El proceso de generación de C# es simple en comparación con el de C y C++, y es más flexible que en Java. No hay archivos de encabezado independientes, ni se requiere que los métodos y los tipos se declaren en un orden determinado. Un archivo de código fuente de C# puede definir cualquier número de clases, estructuras, interfaces y eventos. (10)

Ventajas:

- ✓ 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.
- ✓ Posee capacidades de reflexión.
- ✓ Es flexible en cuanto al orden de definición de las clases y las funciones.
- ✓ Soporta definición de clases dentro de otras.
- ✓ Todos los valores son inicializados antes de ser usados (automáticamente por defecto, o manualmente desde constructores estáticos).

1.4.2. Plataforma .NET

En 1998 un equipo de trabajo de Microsoft Corporation comenzó a trabajar en el proyecto Próxima Generación de Servicios Windows la 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.(11)

Está diseñado para utilizar los servicios web con XML¹² 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.(12)

Entre los lenguajes que se han adherido a esta familia se encuentra el ya mencionado C#. Agrupa además al Microsoft Visual Basic, C++, Java, Pascal, entre otros conformando un grupo de más de veinte.

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.(13)

1.4.3. Framework 3.5

Microsoft .NET Framework 3.5 contiene muchas y nuevas características que se agregan de forma incremental a .NET Framework 2.0 y 3.0 e incluye .NET Framework 2.0 Service Pack 1 y .NET Framework 3.0 Service Pack 1.

NET Compact Framework versión 3.5 amplía la compatibilidad con aplicaciones móviles distribuidas al incorporar la tecnología WCF¹³. También agrega nuevas características de lenguaje como LINQ¹⁴, incluye

¹² Lenguaje de marcado ampliable o extensible (Xtensible Markup Language).

¹³Windows Communication Foundation.

nuevas API¹⁵ basadas en los comentarios de la comunidad y mejora la depuración con herramientas y características de diagnóstico actualizadas.

.NET Framework 3.5 incorpora características mejoradas en áreas concretas de ASP.NET y Visual Web Developer. El avance más significativo es la mejora de la compatibilidad con el desarrollo de sitios web habilitados para AJAX. ASP.NET agrega compatibilidad con el desarrollo de AJAX centrado en el servidor mediante un conjunto de nuevos controles de servidor y nuevas API. Puede habilitar una página ASP.NET 2.0 existente en AJAX agregando un control ScriptManager y un control UpdatePanel, de modo que la página pueda actualizarse sin que sea necesario realizar una actualización de la página completa.

.NET Compact Framework 3.5 admite el generador de perfiles de CLR¹⁶, que sólo estaba disponible con la versión completa de .NET Framework. El generador de perfiles de CLR permite ver el montón administrado de un proceso e investigar el comportamiento del recolector de elementos no utilizados. El generador de perfiles de CLR y su documentación asociada están incluidos en las herramientas avanzadas de .NET Framework 3.5. Admite además la herramienta de configuración, que proporciona información sobre la versión del motor en tiempo de ejecución y funciones administrativas.

Incluye biblioteca ASP.NET AJAX para desarrollar aplicaciones web más eficientes, interactivas y altamente personalizadas que funcionen para todos los navegadores más populares y utilicen las últimas tecnologías y herramientas Web, incluyendo Silverlight y Popfly.(14)

1.4.4. Entorno de desarrollo integrado

Visual Studio 2008

Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#,

¹⁴Conjunto de extensiones de .NET Framework que habilitan la definición de consultas integradas en el lenguaje (Language Integrated Query).

¹⁵ Interfaz de programación de aplicaciones (Application Programming Interface).

¹⁶ CLR: componente de máquina virtual de la plataforma .Net de Microsoft

ASP.NET¹⁷ y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio 2008 ofrece a desarrolladores nuevas herramientas para la fácil creación de aplicaciones conectadas en las últimas plataformas incluyendo web, Windows Vista, Office 2007, SQL Server 2008 y Windows Server 2008. Para la web, se tiene ASP.NET, AJAX y otras tecnologías como Silverlight, WPF. que brindara la posibilidad de crear aplicaciones con rica interfaz de usuario, para poder dar una experiencia de usuario sin precedentes.(15)

Visual Studio 2008 provee un nuevo lenguaje de consultas integrado para el manejo de la información, el cual se llama Microsoft LINQ, que hará la vida más fácil para programadores individuales para poder construir soluciones que analicen y actúen sobre la información.

Visual Studio 2008 ahora permite la creación de soluciones multiplataforma adaptadas para funcionar con las diferentes versiones de .Net Framework: 2.0. (Incluido con Visual Studio 2005), 3.0 (incluido en Windows Vista) y 3.5 (incluido con Visual Studio 2008).

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET. Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.(16)

Características

- ✓ Ampliación del texto y de los botones de las barras de herramientas.
- ✓ Opciones de tamaño de texto para los editores.
- ✓ Personalización de colores para los editores.
- ✓ Personalización de métodos abreviados de teclado.
- ✓ Finalización automática para los métodos y parámetros.

¹⁷Plataforma 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.

1.4.5. Herramientas Case

Enterprise Architect 7.5

Enterprise Architect es una herramienta CASE¹⁸ para el diseño y construcción de sistemas de software. Soporta la especificación de UML 2.0, que describe un lenguaje visual por el cual se pueden definir mapas o modelos de un proyecto.

Enterprise Architect es una herramienta progresiva que cubre todos los aspectos del ciclo de desarrollo, proporcionando una trazabilidad completa desde la fase inicial del diseño a través del despliegue y mantenimiento. También provee soporte para pruebas, mantenimiento y control de cambio.(17)

Algunas de las características claves de Enterprise Architect son:

- ✓ Generar código para el software que está construyendo.
- ✓ Crear elementos del modelo UML para un amplio alcance de objetivos.
- ✓ Ubicar esos elementos en diagramas y paquetes.
- ✓ Crear conectores entre elementos.
- ✓ Documentar los elementos que ha creado.
- ✓ Realizar ingeniería inversa del código existente en varios lenguajes.
- ✓ Importación/Exportación XMI 2.1.¹⁹
- ✓ Modelado basado en el Equipo.
- ✓ Archivos compatibles o modelos basados del repositorio.
- ✓ Control de versiones con cualquier herramienta SCC²⁰.

¹⁸ Ingeniería de software asistida por computadoras (Computer Aided Software Engineering).

¹⁹ Importación/Exportación.

- ✓ Seguridad incorporada y administración de permisos.

Usando Enterprise Architect, puede realizar ingeniería directa y reversa de código C++, C#, Delphi, Java, Python, PHP, VB.NET y clases de Visual Basic, sincronizar códigos y elementos del modelo, diseñar y generar elementos de base de datos. La documentación de alta calidad puede ser rápidamente exportada desde sus modelos en industria estándar formato RTF²¹ e importar a Word para una personalización y presentación final.(18)

Enterprise Architect sustenta todos los diagramas y modelos UML. Puede modelar procesos de negocio, sitios web, interfaces de usuario, redes, configuraciones de hardware, mensajes y más. Estimar el tamaño de su proyecto en esfuerzo de trabajo en horas. Capturar y trazar requisitos, recursos, planes de prueba, solicitudes de cambio y defectos. Desde los conceptos iniciales hasta el mantenimiento y soporte, Enterprise Architect tiene las características que precisa para diseñar y administrar su desarrollo e implementación. (19)

1.4.6. Lenguaje de Modelado

UML 2.0

UML²², es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG²³. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

²⁰Herramienta utilizada para el control de versiones (Source Code Control).

²¹Formato de texto enriquecido (Rich Text Format).

²² Lenguaje Unificado de Modelado (Unified Modeling Language).

²³Objeto de administración de grupos (Object Management Group).

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

1.4.7. Librería

ASP.NET AJAX Control Toolkit:

El ASP.NET AJAX Control Toolkit nace como un proyecto conjunto entre la comunidad de programadores y Microsoft. Está desarrollado en base a ASP.NET AJAX y contiene una serie de controles Web y extendedores con los que se podrá utilizar las avanzadas características de ASP.NET AJAX sin más que un arrastre de ratón. Del mismo modo, con su descarga se disponen de ejemplos de uso, así como del propio código fuente de los controles. Y sobre todo es que es totalmente gratuito.

Se trata de una serie de pequeñas funcionalidades que cualquier webmaster, en su historia de programador, ha utilizado o deseado para sus Webs. La diferencia ahora es que los ejemplos son 100% AJAX y usarlos no exige conocimiento alguno sobre AJAX, JavaScript o XML. Estos controles van desde un simple botón con una alerta asociada, hasta un complejo panel que se puede arrastrar por la pantalla; en ambos casos, mandando y recogiendo información entre el cliente y el servidor sin ningún tipo de recarga de página. Su uso hará que la Web sea mucho más atractiva y simpática al usuario a la par que potente y efectiva. (20)

ASP.NET AJAX, anteriormente llamado Atlas, es un conjunto de extensiones para ASP.NET desarrollado por Microsoft para implementar la funcionalidad de Ajax.

Mediante componentes del lado del cliente y del servidor, ASP.NET AJAX permite al desarrollador crear aplicaciones web en ASP.NET 2.0 que pueden actualizar datos en la página web sin un recarga completa de la misma.

ASP.NET AJAX fue liberado en enero de 2007 después de un largo periodo de pruebas. Fue subsecuentemente incluido con la versión 3.5 del .NET Framework, que fue liberada junto con Visual Studio.(21)

Microsoft AJAX Library es una colección autónoma de clases en JavaScript estandarizadas incluidas con ASP.NET AJAX. Es admitida por la mayoría de los navegadores más populares y puede ser usada para construir aplicaciones web centradas en el cliente que integradas con un proveedor de datos.(20)

1.5. Metodología 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. Pueden ser comparadas con un plan de contingencias en el que se van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además quienes deben participar en el desarrollo de las actividades y qué papel deben de tener. Detallan también la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

RUP

RUP²⁴ constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos, basado íntegramente en Lenguaje Unificado de Modelado como soporte a la metodología. El RUP es un producto de Rational IBM²⁵. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso, que plantea la implementación del proyecto a realizar en iteraciones, con lo cual se pueden definir objetivos por cumplir en cada iteración y así poder ir completando todo el proyecto iteración por iteración. Como parte del enfoque iterativo se encuentra la flexibilidad para acomodarse a nuevos requisitos o a cambios tácticos en los objetivos del negocio.

El desarrollo bajo el Proceso Unificado está centrado en la arquitectura. Este proceso se centra en establecer una arquitectura de un sistema y una arquitectura ejecutable construida como un prototipo evolutivo. Este diseño arquitectónico sirve como una sólida base sobre la cual se puede planificar y manejar el desarrollo de software basado en componentes.

RUP divide el proceso en cuatro fases:

- ✓ Inicio: En esta primera fase se define el ámbito y objetivos del proyecto, así como las funcionalidades y capacidades del producto.

²⁴ Proceso Unificado Racional (Rational Unified Process).

²⁵ Máquina de negocios internacional (International Business Machines).

- ✓ **Elaboración:** En esta fase se define una arquitectura básica y se planifica el proyecto considerando recursos disponibles.
- ✓ **Construcción:** Esta fase proporciona un producto construido junto con la documentación basada en la arquitectura de la línea base.
- ✓ **Transición:** Se libera el producto y se entrega al usuario para un uso real.

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo, los cuales son Modelamiento del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba o Testeo, Instalación, Administración de Configuración y Cambios y Ambiente.

Los elementos de RUP son:

- ✓ **Actividades:** Son los procesos que se llegan a determinar en cada iteración.
- ✓ **Trabajadores:** Son los roles o entes involucrados en cada proceso.
- ✓ **Artefactos:** Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.(22)

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.

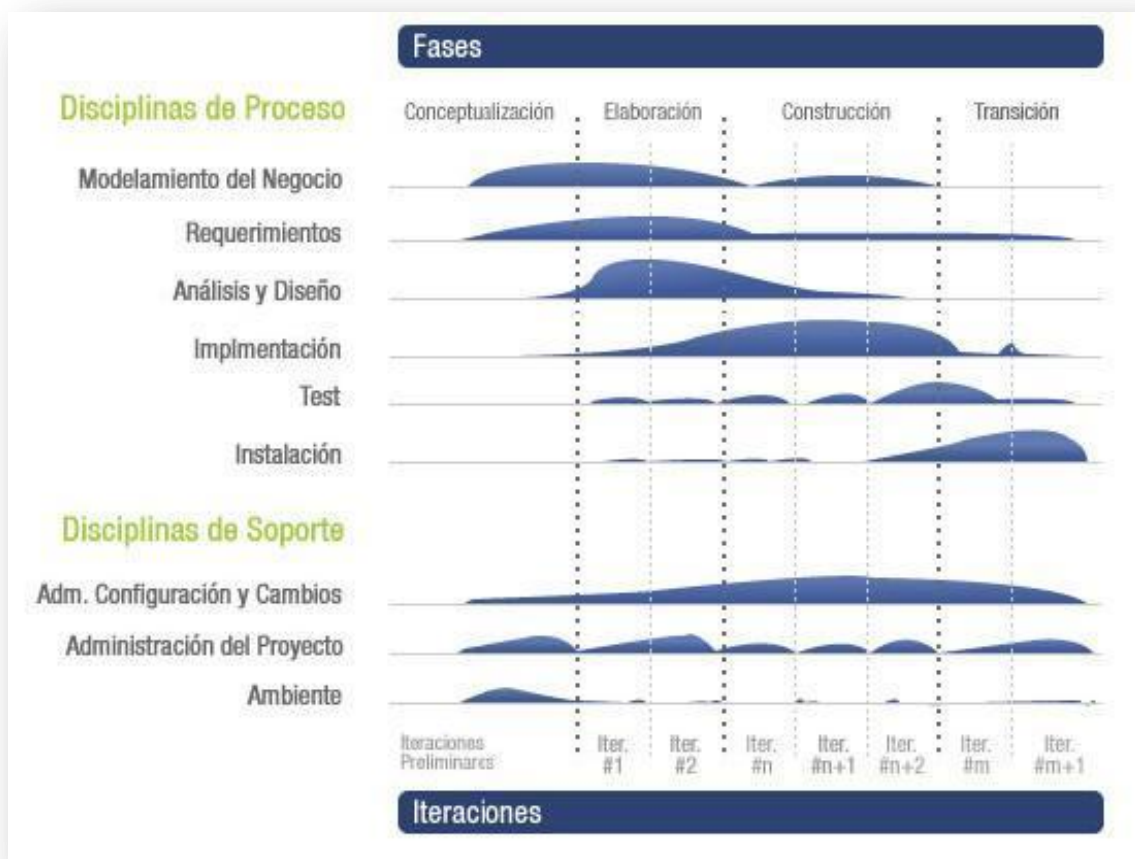


Ilustración 1: Fases e Iteraciones de la Metodología RUP

1.6. Formatos de los documentos a incorporar en la aplicación

Los procesadores de textos brindan una amplia gama de funcionalidades, ya sea tipográficas, idiomáticas u organizativas, con algunas variantes, según el programa de que se disponga. Como regla general, todos pueden trabajar con distintos tipos y tamaños de letra, formato de párrafo y efectos artísticos; además de brindar la posibilidad de intercalar o superponer imágenes u otros objetos gráficos dentro del texto.

Como ocurre con la mayoría de las herramientas informáticas, los trabajos realizados en un procesador de textos pueden ser guardados en forma de archivos, usualmente llamados documentos, así como impresos a través de diferentes medios.

Los procesadores de texto también incorporan desde hace algunos años correctores de ortografía y gramática, así como diccionarios multilingües y de sinónimos que facilitan en gran medida la labor de redacción.

En el sistema operativo Windows se exporta en los siguientes formatos PDF, RTF, XLS, XML, CSV, TXT, HTML, EXCEL, DOC. Mientras que los formatos de textos en Linux los más habituales son:

- ✓ Textos: Estándar de facto de DOC (.doc), formato propietario de MS-Word. Se debe abandonar y utilizar el formato libre: OpenDocument Text (.odt).
- ✓ Documentos electrónicos: Estándar de facto: PDF (.pdf), formato libre de Adobe. Todos los formatos estándar de lenguajes de descripción de página son libres.

A continuación se muestra algunas generalidades, ventajas y desventajas de estos formatos.

PDF

Es multiplataforma, es decir, puede ser presentado por los principales sistemas operativos (Windows, Unix/Linux o Mac), sin que se modifiquen ni el aspecto ni la estructura del documento original. Puede ingerir cualquier combinación de texto, elementos multimedia como vídeos o sonido, elementos de hipertexto como vínculos y marcadores, enlaces y miniaturas de páginas. Es uno de los formatos más extendidos en Internet para el intercambio de documentos. Por ello es muy utilizado por empresas, gobiernos e instituciones educativas.

XML

Es un lenguaje de etiquetado extensible muy simple, pero estricto, que juega un papel fundamental en el intercambio de una gran variedad de datos. Sirve para estructurar, almacenar e intercambiar información. Es la base de los servicios Web, el contenido almacenado en un documento XML se puede transferir fácilmente a través de la red. Los servicios Web XML actúan de forma independiente y además permiten que las aplicaciones compartan información e invoquen funciones de otras aplicaciones independientemente del sistema operativo o la plataforma en que se ejecutan y los dispositivos utilizados para obtener acceso a ellos.

HTML

HTML²⁶, es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.(23)

HTML también es usado para referirse al contenido del tipo de MIME text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores).

Ventajas del HTML:

- ✓ Es el lenguaje de formateo para los navegadores web.
- ✓ Es fácil de entender y utilizar.
- ✓ Su uso es muy extendido.

1.7. Formato de impresión

La aplicación cuenta con diferentes condiciones que le brindan al usuario la oportunidad de hacerle modificaciones al documento antes de imprimir. A las tablas y las imágenes se les podrán cambiar el formato, la alineación, en el sentido que desee el usuario, además de modificar el tamaño. A los textos se les podrá aplicar diferentes estilos.

El usuario para poder modificar algún tipo de error ortográfico que haya detectado debe ir al formulario donde se llenan las plantillas de forma dinámica para modificar el contenido del documento, ya que en la vista de impresión no se realiza este proceso.

²⁶Lenguaje de Marcado de Hipertexto (Hyper Text Markup Language).

1.8. Tecnologías y herramientas a utilizar

Luego del estudio realizado se arriba a la conclusión de que el sistema estará guiado por la metodología de desarrollo RUP, por ser una de las más utilizadas para el diseño, implementación y documentación de sistemas web; además está basada en UML. Por las potencialidades ofrecidas se escoge como herramienta de modelado Enterprise Architect 7.1. Se utilizará el Visual Studio .NET 2008, tecnología ASP.NET –pues se trata de una aplicación web con lenguaje de programación C#. Se incluye la librería ASP.NET AJAX Control Toolkit. La aplicación se sustentará en una Arquitectura Cliente. La aplicación exporta los documentos según el sistema operativo sobre el cual se esté trabajando, en Windows se exportara a PDF, XML, TXT, HTML, DOC, mientras que en Linux se va a exportar a OpenDocument Text (.odt), Open Document Spreadsheet (.ods), Open Document Presentation (.odp).

Como resultado del análisis realizado durante el presente capítulo se concluye que: las herramientas estudiadas relacionadas a los diferentes tipos y funciones de los procesadores de textos en la web, no responden a las necesidades del grupo de Calidad del Centro de Informática Médica, debido a esto se necesita implementar un sistema informático que permita la edición del formato de impresión de los documentos en la web. Se realizó un estudio de las tecnologías actuales y se seleccionaron las más adecuadas para la implementación del sistema.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

En este capítulo se realiza la descripción del sistema a desarrollar, para lograr una mayor claridad y comprensión por parte de los desarrolladores. Los puntos que se abordarán son los siguientes: descripción del modelo de dominio, especificación de los requisitos funcionales y no funcionales, representación de los actores y casos de uso a través de un diagrama de casos de uso del sistema, así como la descripción textual de ellos.

Además de que se tratan los aspectos fundamentales relacionados con el objeto de estudio. Se explican los procesos del negocio sobre los que se basa la organización para su funcionamiento, se describe el flujo actual de los procesos y se realiza un análisis crítico de cómo estos se realizan actualmente -incluyendo toda la información que se maneja, o sea documentos específicos que se procesan-. También se presenta el objeto de automatización.

Se aborda el tema relacionado con la modelación del negocio, siendo este el primer flujo de trabajo durante el proceso de desarrollo de un software, el cual tiene entre sus objetivos comprender los procesos de negocio de la organización, identificar las mejoras potenciales y señalar puntualmente los requerimientos del sistema que va a soportar la organización. Se muestra la especificación de los requisitos de software (los requisitos funcionales y no funcionales), además de la descripción de los actores y casos de uso resultantes del Flujo de trabajo requerimiento.

2.1. Objeto de estudio

2.1.1. Situación Problemática

En el Grupo de Calidad del Centro de Informática Médica (CESIM) se desarrolló una herramienta para la Gestión de la Información de los Expedientes de Proyecto (GIEP 1.0), la cual desempeña diversas tareas relacionadas con la organización de la documentación que existe en los proyectos de la universidad. Actualmente los formularios que se generan se hacen de forma estática, por lo que existen algunos problemas como son:

- ✓ No se puede cambiar la estructura de los documentos del EP que usa la aplicación.

- ✓ El sistema no posee un mecanismo de razonamiento para que en dependencia de la información contenida en estas plantillas XML la aplicación genere dinámicamente un formulario para un XML determinado.
- ✓ Si los documentos XML son modificados la aplicación no es compatible con la mayor parte de estos.

Con el uso de la herramienta que se propone en este trabajo se garantiza la automatización de los procesos y la solución real a los problemas que existen.

2.2. Objeto de automatización

Resulta de vital importancia que GIEP garantice que los cambios ocurridos en los XML no afecten el buen funcionamiento de la aplicación por lo que se desea obtener el siguiente proceso: Generar plantillas del sistema para la Gestión de información del expediente de proyecto.

2.3. Propuesta del sistema

Se propone la implementación de un componente que genere documentos en diferentes extensiones dinámicos basados en documentos XML, para ello la herramienta contará con un área de diseño donde se podrá definir el formato, además una vez creado el documento pueden diseñarse varias versiones sin necesidad de cambiar la versión anterior, con el objetivo de mantener los cambios realizados al mismo.

2.4. Descripción del Modelo del dominio

El Modelo de Dominio es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema. Representa conceptos del mundo real, no de los componentes de software. Una clase conceptual puede ser una idea o un objeto físico (símbolo, definición y extensión).

El modelo de dominio se representa en UML con un Diagrama de Clases en los que se muestra:

- ✓ Conceptos u objetos del dominio del problema: clases conceptuales.
- ✓ Asociaciones entre las clases conceptuales.
- ✓ Atributos de la clase conceptuales.

En el Modelo de Dominio no se muestra comportamiento. Las clases conceptuales pueden tener atributos pero no métodos. Cualquiera sea la solución de casos de uso que se haya elegido, los conceptos e ideas propias del dominio del problema son las mismas; un mismo modelo de dominio contempla cualquiera de las soluciones analizadas. El modelo de dominio es global, es decir se realiza para todos los casos de uso y no para uno en particular.

El modelo de dominio se realiza si no se logran determinar los procesos de negocio con fronteras bien establecidas donde se logren ver claramente, quienes son las personas que realizan cada proceso de negocio, quienes son los beneficiados con cada uno de estos procesos, pero además quienes son las personas que desarrollan las actividades en cada uno de estos procesos.(24)

Al mismo tiempo se logran identificar los siguientes conceptos:

GIEP: Herramienta para la gestión de la información del expediente de proyecto.

Plantilla XML: Es un archivo o fichero utilizado por la herramienta GIEP para la elaboración de un estándar, que carga y guarda información.

Documento: Es el artefacto que se genera en el proceso de desarrollo de software que se recoge en el Expediente de Proyecto.

PDF: Formato en la que la aplicación va a exportar y luego a imprimir.

El modelo de dominio se describe mediante diagramas de UML, especialmente mediante diagramas de clases. En la ilustración (2) se representa el modelo de dominio realizado.

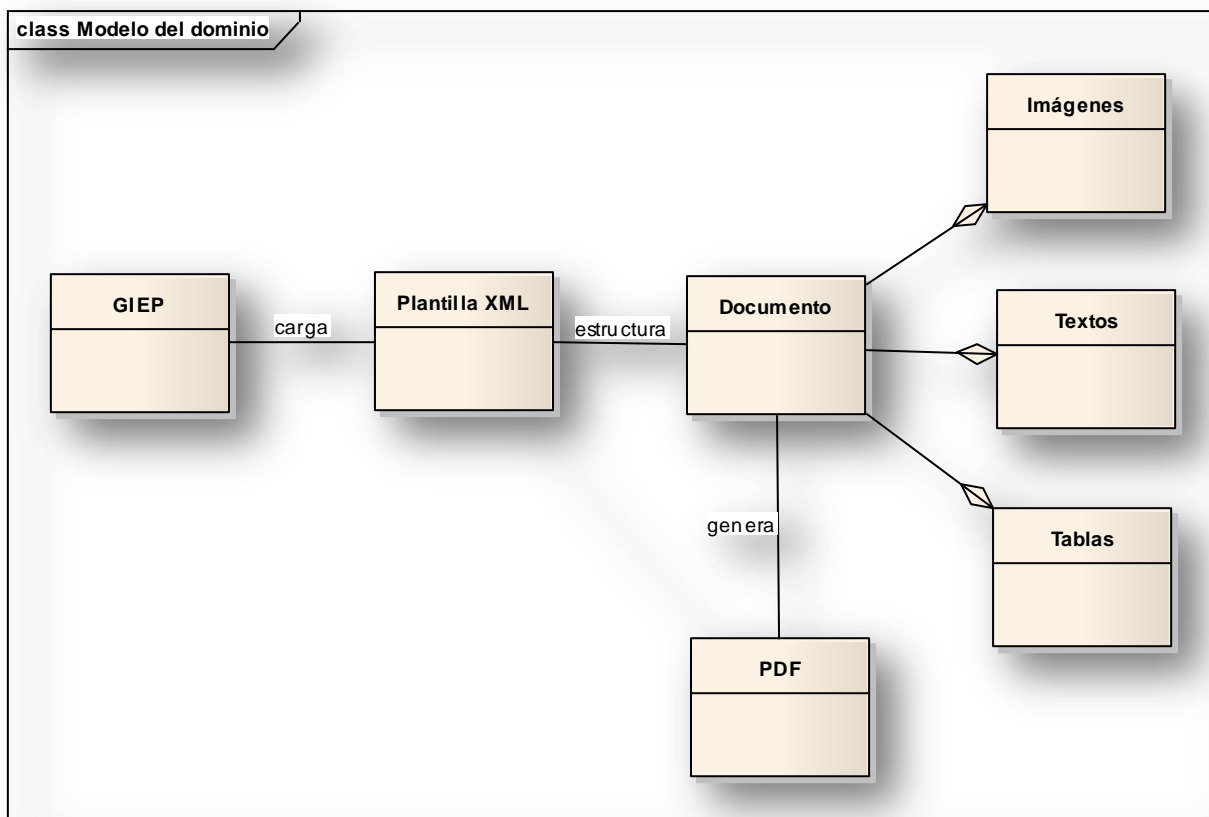


Ilustración 2: Diagrama del modelo del Dominio

2.5. Especificación de Requisitos de Software

El objetivo fundamental del flujo de trabajo es guiar el desarrollo al sistema correcto, lo que se consigue mediante una descripción de los requisitos de la aplicación, es decir, condiciones o capacidades que debe cumplir. Los requisitos deben ser especificados por escrito, descritos como una característica del sistema a construir y posibles de probar o verificar.(25)

2.6. Requerimientos Funcionales del Sistema

Son capacidades o condiciones que el sistema debe cumplir sin alterar la funcionalidad del producto, esto quiere decir que se mantienen invariables sin importar con qué propiedades o cualidades se relacionan. Su

principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema.(26)

A partir del modelo de dominio representado anteriormente, se alcanzó la visión necesaria del objeto a automatizar y quedaron determinadas las funcionalidades. Con ellos, se pretende determinar de manera clara y concisa lo que debe hacer el sistema. A continuación se muestran los requisitos que debe cumplir el sistema:

RF1: Editar vista previa

- ✓ **RF1.1:**Cargar plantilla XML
- ✓ **RF1.2:**Ajustar tamaño de página
- ✓ **RF1.3:** Ajustar tamaño de imagen
- ✓ **RF1.4:** Diseñar el estilo de página
- ✓ **RF1.5:** Editar información
- ✓ **RF1.6:** Ver vista de la página

RF2: Mostrar vista previa

RF3: Deserializar la información

RF4: Imprimir documentos

- ✓ **RF4.1:**Imprimir documento en formato PDF
- ✓ **RF4.2:**Establecer margen al documento
- ✓ **RF4.3:**Establecer formato de impresión a los documentos

RF5: Exportar documentos

- ✓ **RF5.1:**Exportar documento a PDF
- ✓ **RF5.2:**Exportar documento XML de Word

- ✓ **RF5.3:**Exportar documento a HTML
- ✓ **RF5.4:**Exportar Documento de Word (.doc)
- ✓ **RF5.5:**Exportar documento a .odt

RF6: Guardar Configuración

- ✓ **RF6.1:**Registrar usuario que realizó la personalización
- ✓ **RF6.1.1:**Registrar Fecha y hora del cambio
- ✓ **RF6.2:**Cargar
- ✓ **RF6.3:**Eliminar
- ✓ **RF6.4:**Guardar
- ✓ **RF6.5:**Editar
- ✓ **RF6.6:**Crear Nueva

2.7. Requerimientos No Funcionales del Sistema

Los requisitos no funcionales son características requeridas del sistema, del proceso de desarrollo, del servicio prestado o de cualquier otro aspecto del desarrollo, que señala una restricción del mismo, que se aplican de manera general como un todo, más que a rasgos particulares del mismo.

Estos requerimientos son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.(27)

Requerimientos de Usabilidad

Nota: Se usa el prefijo **RNU** en su nomenclatura

RNU 1 La interfaz debe ser configurable para lograr la comodidad del usuario.

RNU 2 El usuario debe poder configurar el sistema, sin necesidad de acceder manualmente a los ficheros de configuración.

RNU 3 El sistema debe mostrar mensajes al usuario, que le ayuden a llevar a cabo la tarea que realiza.

RNU 4 Se debe hacer uso de botones, con imágenes que indiquen de modo intuitivo la función que realizan.

RNU 5 Los controles visuales deben mostrar mensajes que indiquen su función.

RNU 6 La configuración predeterminada debe ser cómoda y amigable.

RNU 7 Los parámetros de funcionamiento del módulo deben ser configurables.

Requerimientos de Fiabilidad

Nota: Se usa el prefijo **RNF** en su nomenclatura.

RNF 7 El sistema debe estar disponible de forma permanente.

Requerimientos de Diseño e Implementación

Nota: Se usa el prefijo **RNDI** en su nomenclatura.

RNDI 8 Debe ser implementado en C# ajustándose a las funcionalidades del Framework 3.5.

RNDI 9 Resolución de 1024 x 768.

Requerimientos de Interfaz de usuario

Nota: Se usa el prefijo **RNIU** en su nomenclatura.

RNIU 10 Se requiere que el Cliente tenga una interfaz gráfica que permita la interacción con el usuario.

Requerimientos de Funcionamiento

Nota: Se usa el prefijo **RNFO** en su nomenclatura.

RNFO 11 Se debe tener instalado .NET Framework 3.5 o superior.

RNFO 12 Usar como sistema operativo Windows XP SP2 o superior: Determinado por los requerimientos del .NET Framework 3.5.

Requerimientos de Seguridad

*Nota: Se usa el prefijo **RNS** en su nomenclatura.*

RNS 13 Se garantizará la seguridad del Sistema mediante la autenticación de usuarios.

RNS 14 Se registrarán las acciones llevadas a cabo en el sistema.

2.8. Definición de los Casos de Uso del Sistema

Los casos de uso son el componente clave del modelado. Su propósito es ilustrar como un sistema permite a un actor cumplir una meta, ilustrando todos los posibles caminos apropiados que ellos pueden tomar para cumplirla, así como las situaciones que podrían hacerlo fallar.

El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requerimientos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El mismo contiene actores, casos de uso y sus relaciones y constituye una entrada para el análisis, el diseño y las pruebas.(25)

2.8.1. Casos de Uso del Sistema

CUS 1: Editar vista previa

CUS 2: Mostrar vista previa

CUS 3: Imprimir documentos

CUS 4: Exportar documentos

CUS 5: Guardar configuración

2.8.2. Diagrama de Casos de Uso del Sistema

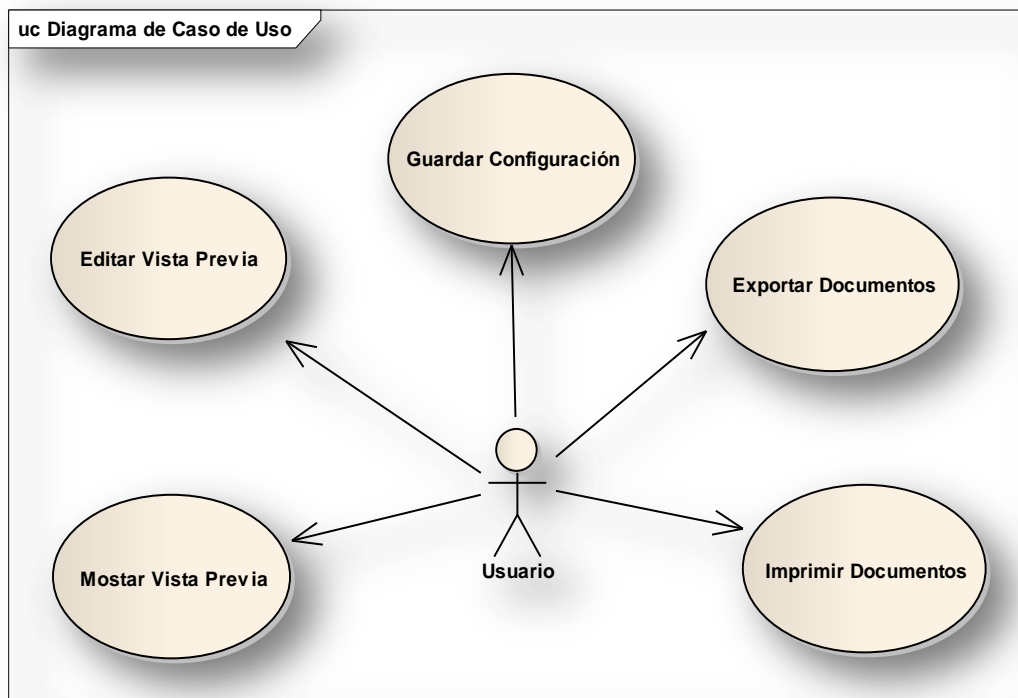


Ilustración 2: Diagrama de Casos de Uso del Sistema

2.9. Definición de los actores

Los actores representan terceros fuera del sistema que colaboran con el sistema. Al identificar los actores del sistema se identifica el entorno externo del sistema. No son parte de él, pueden intercambiar información con él, ser un recipiente pasivo de información y representar el rol que juega una o varias personas, un equipo o un sistema.

Actores	Justificación
Usuario	Es el encargado del manejo de toda la información gestionada durante el proceso.

Tabla 1: Descripción del Actor del Sistema

2.10. Descripción expandida de los Casos de Uso

De acuerdo a las nuevas funcionalidades que se le agregarán al Sistema para la Gestión de Información del Expediente de Proyecto quedan conformados los siguientes Casos de Uso:

Caso de Uso Editar Vista Previa

Caso de uso	
CUS 1	Editar Vista Previa
Propósito	Mostrar una vista al usuario para que este realice las acciones que desee.
Actores: Usuario	
Resumen: El caso de uso se inicia cuando el usuario selecciona la plantilla, el sistema carga esta plantilla XML. El caso de uso termina cuando el usuario termina de editar la información.	
Referencias	RF1 ,RF1.1 ,RF1.2 , RF1.3 , RF1.4 , RF1.5 ,RF1.6
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. Selecciona la opción editar el documento. 4. Selecciona una de las opciones 	<ol style="list-style-type: none"> 2. El sistema visualiza el editor. 3. El sistema brinda la posibilidad varias acciones tales como: <ul style="list-style-type: none"> • Justificar textos • Modificar colores • Modificar tamaño de las tablas e

	<p>imágenes.</p> <p>5. El sistema realiza la opción seleccionada.</p>
<p>El Caso de Uso termina</p>	

Tabla 2: Descripción del CU Editar Vista Previa

Caso de Uso Mostrar Vista Previa

<p>Caso de uso</p>	
<p>CUS 2</p>	<p>Mostrar Vista Previa</p>
<p>Propósito</p>	<p>Mostrar vista previa</p>
<p>Actores: Usuario</p>	
<p>Resumen: El caso de uso se inicia cuando el usuario accede a la opción mostrar vista previa, el sistema muestra la pantalla para que el usuario vea la información pero sin posibilidades de modificar. El caso de uso culmina cuando el usuario ve la pantalla donde se muestra la vista.</p>	
<p>Referencias</p>	<p>RF2</p>
<p>Flujo Normal de Eventos</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
<p>1. Selecciona la opción mostrar vista previa.</p>	<p>2. Muestra la pantalla con la información del contenido del documento.</p>

El Caso de Uso termina

Tabla 3: Descripción del CU Mostrar Vista Previa

Caso de Uso Imprimir Documentos

Caso de uso	
CUS 3	Imprimir Documentos
Propósito	Darle formato a los documentos a imprimir.
Actores: Usuario	
Resumen: El caso de uso se inicializa cuando el usuario decide imprimir el documento ya editado y con el formato de impresión correcto permitiendo guardar la información o versionar el documento finalizando el caso de uso.	
Referencias	RF4, RF4.1, RF4.2,RF4.3
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1 El usuario selecciona el botón "imprimir" .	2. El sistema muestra la opción de imprimir.
El Caso de Uso termina	

Tabla 4: Descripción del CU Imprimir Documentos

Caso de Uso Exportar Documentos

Caso de uso	
CUS 4	Exportar documentos
Propósito	Darle formato a los documentos y guardarlos.
Actores: Usuario	
Resumen: El caso de uso se inicializa cuando el usuario decide guardar el documento ya editado y con la extensión especificada, permitiendo guardar la información o versionar el documento finalizando el caso de uso.	
Referencias	RF5, RF5.1, RF5.2, RF5.3, RF5.4, RF5.5
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. El usuario selecciona el botón “Exportar”. 3. Selecciona el formato que desea y presiona el botón “aceptar”. 	<ol style="list-style-type: none"> 2. Muestra una ventana para seleccionar el formato a exportar. 4. Exporta el formato seleccionado.
El Caso de Uso termina	
Flujo Alternativo	
3.1 Selecciona la opción “cancelar” .	3.2 Muestra la vista anterior.

El Caso de Uso termina	

Tabla 5: Descripción del CU Exportar

Caso de Uso Guardar Configuración

Caso de uso	
CUS 5	Guardar Configuración
Propósito	Gestionar las configuraciones personalizadas.
Actores: Usuario	
Resumen: El caso de uso se inicia cuando el usuario decide guardar su configuración personal, el sistema le brinda la posibilidad de realizar algunas opciones, además de registrar fecha y hora del cambio. El caso de uso termina cuando el usuario termine de realizar su configuración personal.	
Referencias	RF6, RF6.1, RF6.1.1, RF6.2, RF6.3, RF6.4, RF6.5
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El usuario selecciona el botón “ guardar configuración ”.	2. El sistema guarda la configuración y muestra un mensaje “se guardó la configuración satisfactoriamente”.

El Caso de Uso termina

Tabla6: Descripción del CU Guardar Configuración

En este capítulo se establecieron las características principales del sistema así como el proceso para su funcionamiento. Se muestra el modelo del dominio, conjuntamente con la especificación de los requisitos funcionales y no funcionales. Además se elaboró el modelo de casos de uso del sistema.

CAPÍTULO 3. DISEÑO DEL SISTEMA

Como resultado del flujo de trabajo de requisitos se obtiene una vista externa del sistema, que en el lenguaje del cliente, describe lo que se espera de él a través del Diagrama de casos de uso. A partir de aquí se debe profundizar en los casos de usos detallándolos de manera que permitan reflejar una vista interna del sistema descrita con el lenguaje de los desarrolladores. En esta vista interna se especifican mejor los casos de uso y se determinan las clases necesarias para llevar a cabo las funcionalidades en ellos contenidos. Este proceso se desarrolla fundamentalmente dentro de la fase de elaboración y se corresponde principalmente con el flujo de trabajo de análisis y diseño.

En el presente capítulo se abordarán los aspectos relacionados con este flujo de trabajo; incluyendo, los diagramas de clases del diseño, diagramas de secuencia, y descripción de las clases de diseño.

3.1. Arquitectura

3.1.1. Patrones Arquitectónicos usados

Patrón arquitectónico Modelo-Vista-Controlador

El patrón MVC²⁷ afronta una de las consecuencias de los cambios frecuentes en los soportes de hardware y de añadir nuevas funcionalidades, que es la de constantes modificaciones en la interfaz de usuario. Se separa los módulos en 3 componentes fundamentales, el Modelo, la Vista y el Controlador. El Modelo es inmutable a los cambios en la interfaz de usuario, se encarga de toda la gestión interna del negocio, la Vista se encarga de presentar los datos y de interactuar con el usuario, mientras que el controlador maneja todos los cambios de estado de la Vista. Mediante el uso de este patrón el Sistema de Gestión de la Información del Expediente de proyecto es capaz de utilizar diferentes interfaces de usuario manteniendo la misma lógica de negocio.(28)

Modelo: Es la representación específica de la información con la cual el sistema opera, también puede operar con más datos no relativos a la presentación.

²⁷Modelo-Vista-Controlador.

Vista: Presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.(29)

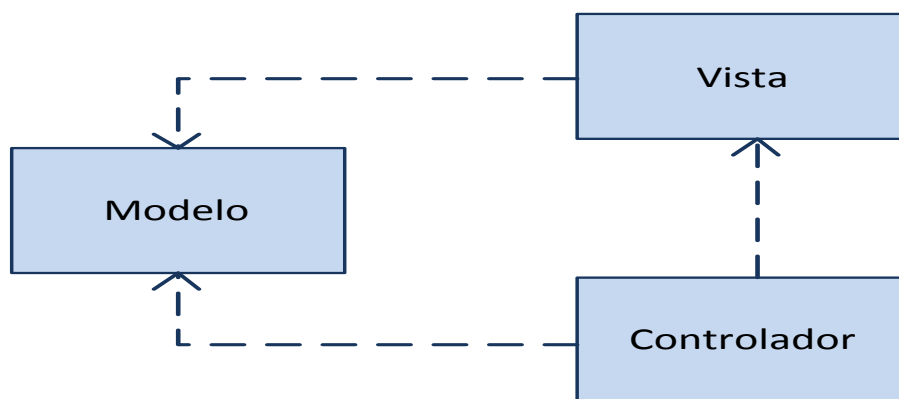


Ilustración 4: Patrón Modelo-Vista-Controlador

3.2. Diseño

3.2.1. Fundamentación del uso de patrones

Para llevar a cabo un buen diseño se han definido una serie de patrones. Los patrones de diseño de software constituyen un conjunto de principios generales y expresiones que ayudan a desarrollar software. Los patrones GRASP²⁸ describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Dentro de este grupo se identifican 5 patrones fundamentales experto, creador, alta cohesión, bajo acoplamiento y el controlador.

²⁸ Patrones Generales de Software para Asignación de Responsabilidades (General Responsibility Assignment Software Patterns).

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.

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, o usa directamente las instancias creadas del objeto, o almacena o maneja varias instancias de la clase.

Alta cohesión: Se dice que 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

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.

En los diagramas de clases elaborados se aplican dichos patrones, se utilizan a fin de distribuir responsabilidades en las mismas, y establecer sus relaciones, tratando de que no estén muy sobrecargadas de funcionalidades ni exista mucha dependencia entre ellas.

3.2.2. Definición de elementos de Diseño

El sistema se encuentra dividido en tres capas bien definidas, que son la presentación, la lógica del negocio y la capa de acceso a datos

Presentación: Incluye las clases interfaces, a través de las cuales se establece la comunicación con los usuarios, y para dar cumplimiento a los pedidos de estos se invocarán los métodos de la lógica del negocio.

Lógica del negocio: Incluye las clases controladoras, así como las entidades del negocio, que representan objetos que van a ser manejados o consumidos por toda la aplicación.

Acceso a datos: Contiene clases que interactúan con la base de datos y permiten, utilizando los procedimientos almacenados generados, realizar todas las operaciones con la base de datos de forma transparente para la capa de negocio.

3.3. 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. El diagrama de clases de diseño representa la parte estática del sistema. Contiene las clases del diseño y sus relaciones. A continuación se presentarán los diagramas de clases del diseño de los casos de uso Editar vista previa e Imprimir .El resto de los diagramas de clases del diseño se podrán encontrar en el Anexo 1.(30)

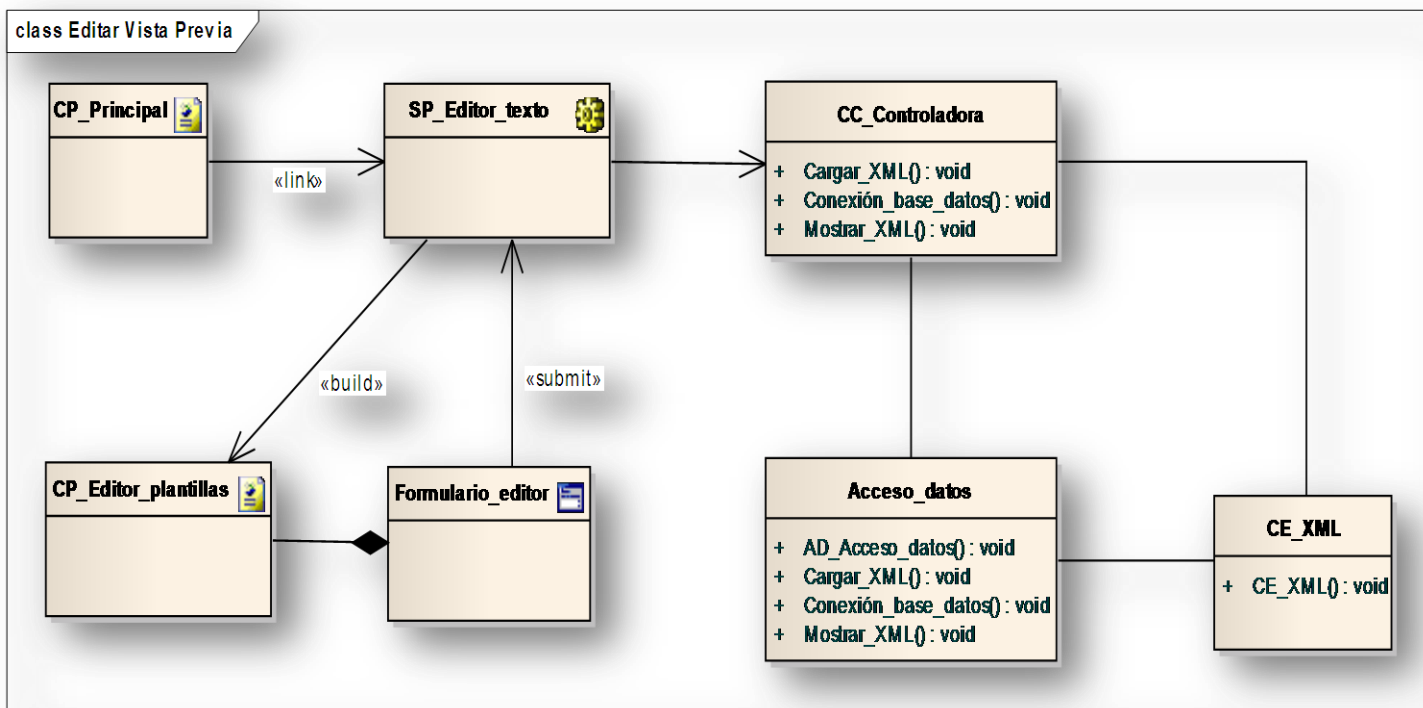


Ilustración 6: Diagrama de clase del diseño CU- Editar Vista Previa

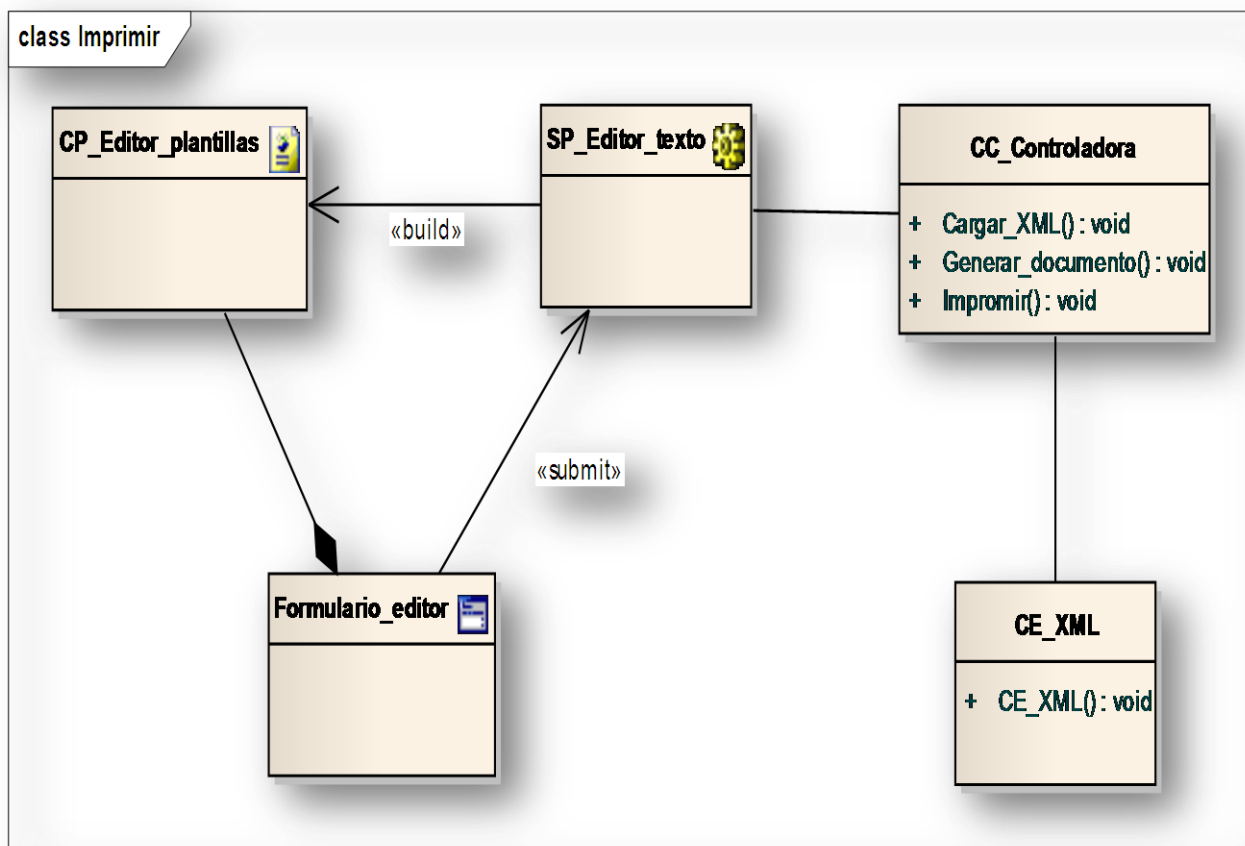


Ilustración 7: Diagrama de clase del diseño CU- Imprimir Documentos

3.3.1. Diagramas de Interacción (Secuencia)

Se presentarán los diagramas de interacción de los Caso de Uso Editar vista previa e Imprimir, dividido por escenarios, para lograr una mejor comprensión. El resto de los diagramas de interacción se podrán encontrar en el Anexo 2.

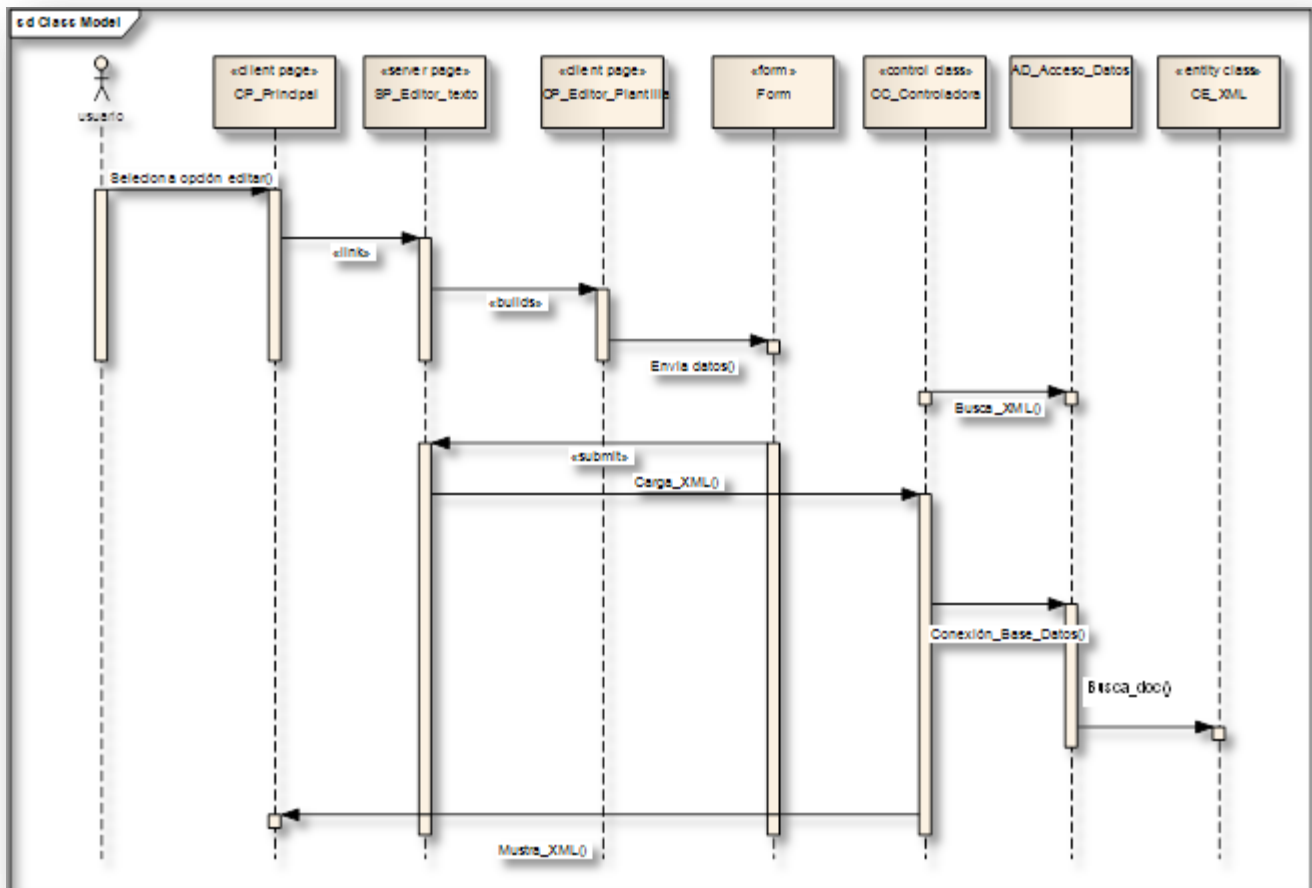


Ilustración 8: Diagrama de Secuencia del CU-Editar Vista Previa

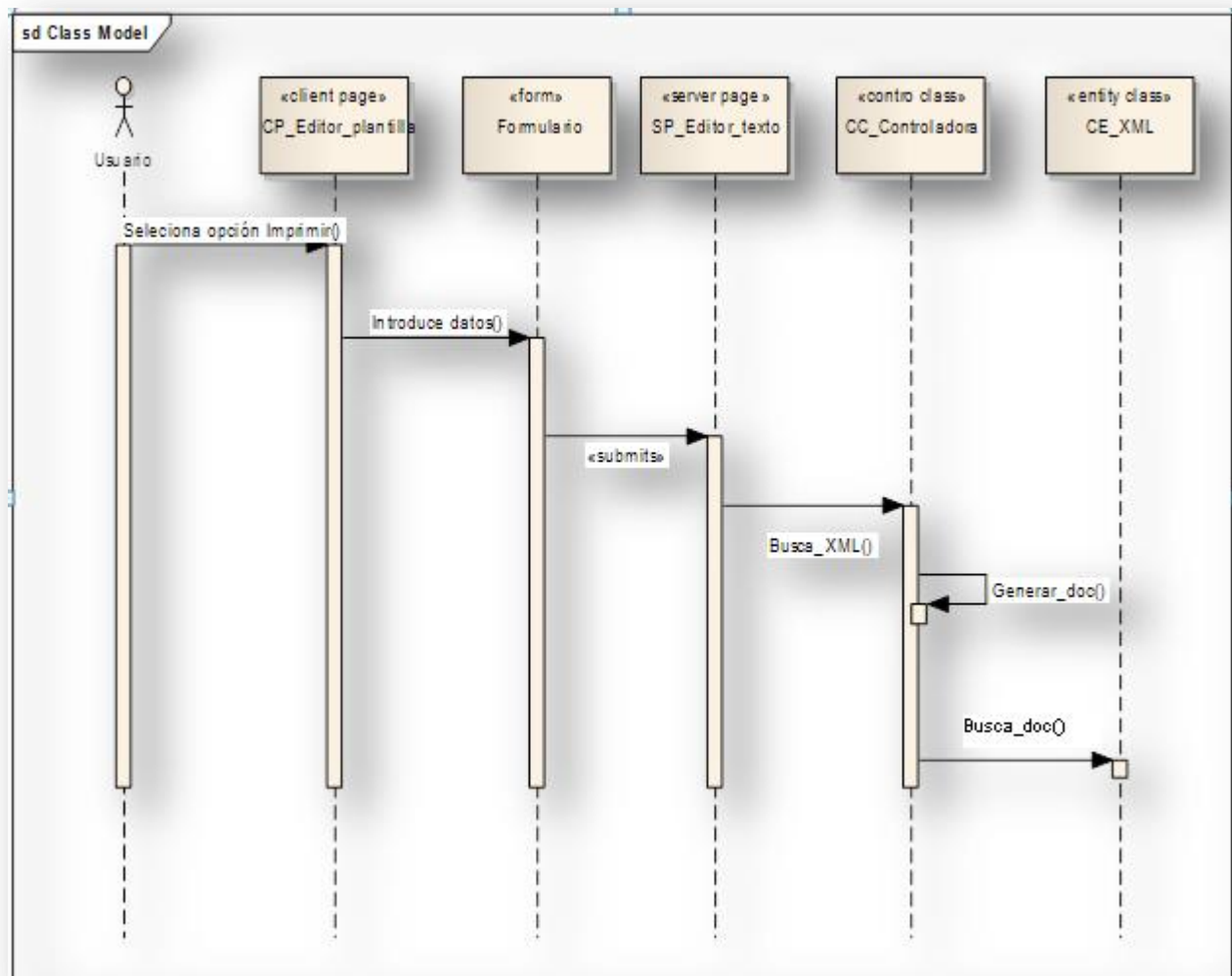


Ilustración 9: Diagrama de Secuencia del CU-Imprimir Documentos

3.4. Descripción de las clases del diseño

Se presentarán las descripciones de las clases de los casos de uso de los cuales se expuso el diagrama.

NOMBRE:	Editor0	
Tipo de Clase:	Controldora	
Método	Tipo	
Deserializar	void	
Principal	void	
Construir	void	
Llenar_Editor	void	
Vista_Previa	Botón	
GenerateMsWordDoc	void	
GenaratePDF	void	
BusquedaElementos	void	
BusquedaXAtributos	string	
Exportar	void	
Editar	void	
Descripción:	Esta clase se encuentran todos los métodos.	

Tabla 2 Descripción de la clase Editor0

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. La arquitectura definida está orientada al logro de una base sólida para la construcción del sistema. Fueron obtenidos los artefactos correspondientes a los flujos de trabajo desarrollados acorde a la metodología utilizada.

CAPÍTULO 4. IMPLEMENTACIÓN

En este capítulo quedan contenidos todos los detalles referentes a la implementación del sistema, mostrando de esta forma los diagramas de componentes dando una vista detallada de cada uno de los paquetes en que se ha dividido y los múltiples elementos físicos y archivos que conforman la aplicación con el objetivo de lograr una mayor claridad y comprensión del modelo, seguidamente el diagrama de despliegue que muestra la distribución del sistema en los diferentes elementos de hardware que le darán soporte.

4.1. Modelo de implementación

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

4.2. Diagrama de despliegue

El diagrama de despliegue es utilizado para modelar el hardware empleado en las implementaciones de sistemas y las relaciones entre sus componentes. El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo.(31)

A continuación se explican los recursos presentes en cada nodo del diagrama:

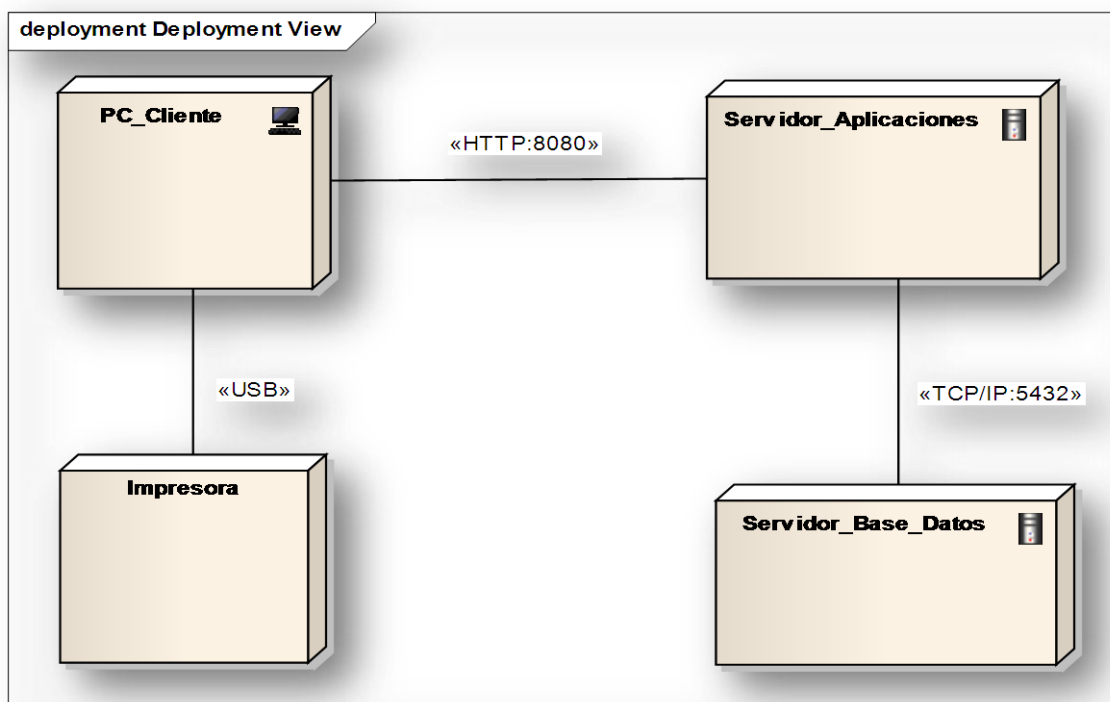


Ilustración 10: Diagrama de Despliegue

4.3. Componentes

Un componente es una parte física y reemplazable de un sistema que se conforma con un conjunto de interfaces y proporciona la realización de dicho conjunto. Representa una unidad de código (fuente, binario o ejecutable) que permite mostrar las dependencias en tiempo de compilación y ejecución. Las instancias de componentes de software muestran unidades de software en tiempo de ejecución y generalmente ayudan a identificar sus dependencias y su localización en nodos.

4.4. Diagrama de Componentes

Un diagrama de Componentes ilustra los fragmentos de software que conformarán un sistema. Un diagrama de componentes tiene un nivel de abstracción más elevado que un diagrama de clase, usualmente un componente se implementa por una o más clases u objetos en tiempo de ejecución. Estos son bloques de construcción, como así eventualmente un componente puede comprender una gran porción de un sistema.

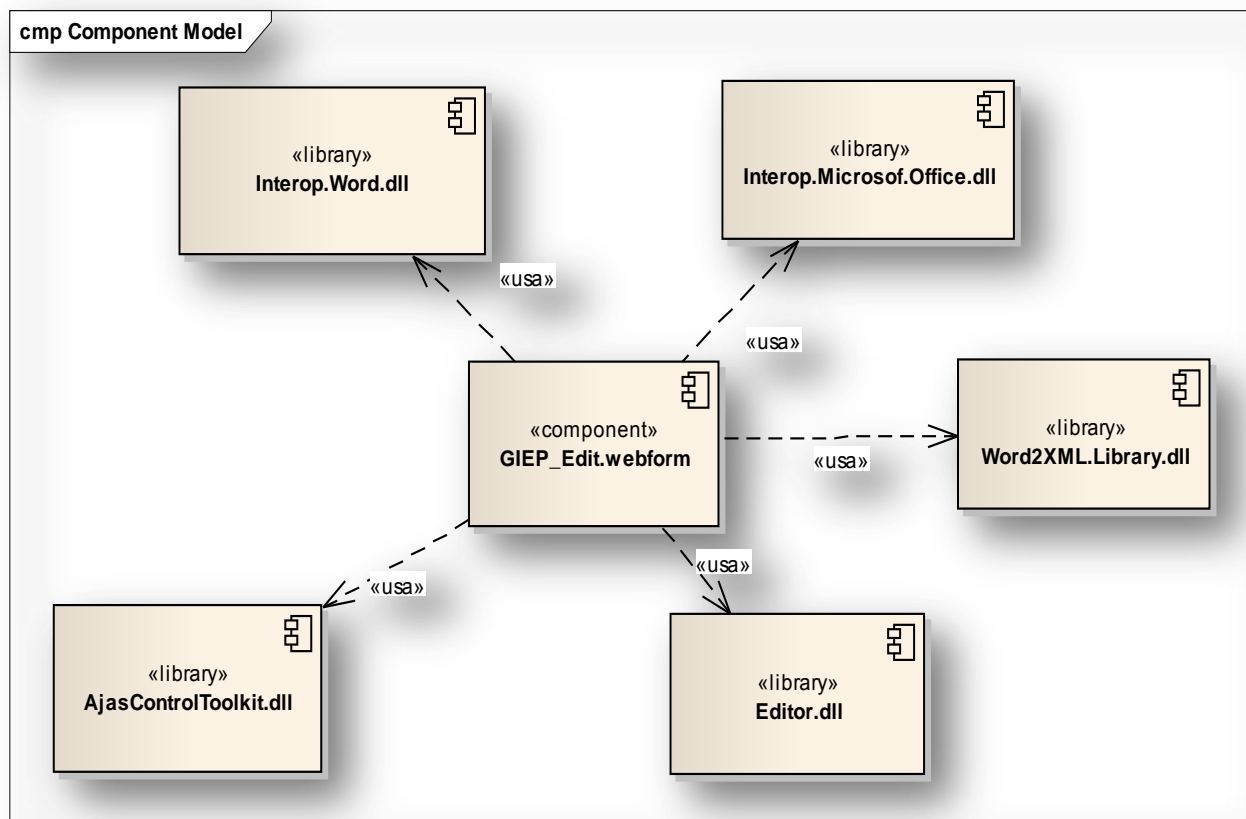


Ilustración 11: Diagrama de Componentes

4.5. Tratamiento de errores

El sistema cuenta con varios mecanismos para el tratamiento errores. El uso de ASP.NET permite usar controles predefinidos que validan los datos de entradas definidos para el usuario sin necesidad de realizar peticiones al servidor, sumándose a esta ventaja el hecho de poder ser utilizados sin necesidad de incluir líneas de código.

La interfaz Web está diseñada para que el usuario realice sus acciones entrando al sistema solo los campos indispensables, ahorrando tiempo al usuario y evitando acciones innecesarias que puedan provocar

errores. Para cumplir este objetivo se cuenta con elementos configurables como campos de selección, listas desplegables con varias opciones y campos autogenerados por el sistema de forma transparente al usuario.

El sistema consta con funciones de validación y captación de errores, informando al usuario en momentos importantes sobre la realización de acciones incorrectas y la forma de corregir dicha acción. El sistema, junto al manejo de excepciones, cuenta con la posibilidad de enviar mensajes explícitos al usuario informando sobre la realización exitosa o cancelación de una acción determinada, evitando siempre abusar de los mensajes de información.

En este capítulo, se obtuvieron los diagramas de componentes y de despliegue, mostrándose como se encuentra dividido por paquetes de clase el desarrollo del componente agrupando clases a fines según la capa a la que pertenecen, presentación y negocio. Se especifica las relaciones que existen entre los componentes que indican el grado de interrelación que existen entre ellos, así como el tratamiento de errores.

CONCLUSIONES

Una vez concluida la investigación se ha dado cumplimiento a los objetivos planteados obteniéndose los resultados que a continuación se mencionan:

Se realizó un análisis de las tendencias actuales de las herramientas y tecnologías lográndose seleccionar las más adecuadas para la implementación del sistema teniendo en cuenta sus particularidades.

Se generó la documentación pertinente para la continuación del desarrollo de la aplicación y futuras versiones.

Se definieron las necesidades del funcionamiento de la aplicación así como los artefactos correspondientes a los flujos de trabajo definidos por la metodología seleccionada.

Se desarrolló un componente informático para la edición del formato de impresión de los documentos generados en la variante Web del sistema de Gestión de la Información del Expediente de Proyecto.

RECOMENDACIONES

Para futuras investigaciones, se recomienda:

- ✓ Agregarles nuevas extensiones cuando la aplicación va a exportar tales como HTML y TXT.
- ✓ Implementar una versión para sistemas operativos libres, aprovechando las potencialidades de la plataforma MONO.

REFERENCIAS BIBLIOGRÁFICAS

1. [En línea] http://calisoft.uci.cu/index.php?view=article&catid=10:art&id=17:expediente-de-proyecto&option=com_content&Itemid=24.
2. [En línea] www.alegsa.com.ar/aplicacion.php..
3. [En línea] www.alegsa.com.ar/procesador%20de%20textos.php.
4. [En línea] <http://www.programacion.org>.
5. Janium.com. *Aplicaciones basadas en Web*. [En línea] 2009. [En línea] 2010. <http://www.janium.com/page2/page1/page6/page7/page7.html>..
6. Masternewmedia.org. *Beneficios De Las Aplicaciones Basadas En Web Y El Anuncio De Microsoft De La Era "En Vivo"*. [En línea] 2005. Be http://www.masternewmedia.org/es/aplicaciones_web/temas_de_aplicaciones_web/Beneficios_De_Las_Aplicaciones..
7. [En línea] <http://www.Masternewmedia.org>.
8. [En línea] <http://www.tecnologiadiaria.com/2009/11/crear-documentos-pdf-acrobat-buzzword.html>..
9. [En línea] <http://www.desarrolloweb.com/articulos/1681.php>..
10. [En línea] <http://decsai.ugr.es/~cb/CSharp/lenguaje/intro.xml>.
11. [En línea] 16- Microsoft Corporation. (2010). Recuperado el 14 de 02 de 2010, de MSDN Visual Studio: <http://msdn.microsoft.com/es-es/library/aa288436%28VS.71%29.aspx>.
12. Subgurim.NET. [En línea] <http://www.subgurim.net/Articulos/asp-net-general/3/por-que-asp-net.aspx>.
13. [En línea] <http://www.info-ab.uclm.es/asignaturas/42580/introdotnet.pdf> .
14. [En línea] <http://social.msdn.microsoft.com/Search/es-es?query=Remoting>.
15. [En línea] <http://social.msdn.microsoft.com>

16. [En línea] <http://msdn.microsoft.com/es-es/library/bb822049%28v=vs.90%29.aspx..>
17. [En línea] <http://msdn.microsoft.com/en-us/vstudio..>
18. Guía de Usuario de Enterprise Architect. [En línea] 05 de Enero de 2011. <http://www.sparxsystems.com.ar>.
19. **Enterprise Architect - Herramienta de diseño UML.** Sparxsystems.com . [En línea] <http://www.sparxsystems.com.ar/products/ea.html..>
20. [En línea] [http://ingenierioscomputacionipla.wordpress.com/2008/09/12/enterprise-architect-71-modelado-uml/..](http://ingenierioscomputacionipla.wordpress.com/2008/09/12/enterprise-architect-71-modelado-uml/)
21. [En línea] Microsoft Corporation. <http://support.microsoft.com/kb/930858/es>.
22. **Pressman, Roger.** *Un Enfoque Práctico.*
23. [En línea] <http://msdn.microsoft.com/es-es/library/fa1h9d0d%28v=vs.80%29.aspx..>
24. [En línea] Modelo de Dominio. [Online] [Citado: Febrero 01,2010] Disponible en http://iie.fing.edu.uy/ense/assign/desasoft/practico/hoja8/ejemplos_clase2.pdf.
25. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* La Habana : Felix Varela, 2004.
26. [En línea] Requerimientos. [Online] [Citado: Febrero 01,2010] Disponible en http://iie.fing.edu.uy/ense/assign/desasoft/practico/hoja8/ejemplos_clase2.pdf.
27. [En línea] http://www.procuraduria.gov.co/descargas/contratacion_2006/licitaciones/licitacion112006bid_Anexo4_4Requerimientosfuncionalesynofuncionalesparte_2.pdf.
28. **Sandó, Lesther Delgado Pérez y José Rolando Pérez.** *Herramienta para la revisión y seguimiento de errores de la documentación generada en los Proyectos de la Facultad 7.*

-
29. **Froufe, Agustín. 2005. ARQUITECTURA Modelo/Vista/Controlador.** [En línea] 2005. http://www.ulpgc.es/otros/tutoriales/java/Apendice/arq_mvc.html..
30. **Autores: Amarilis Peña y Yudisleydis Valencia.** *Trabajo de Diploma Herramienta para automatizar el proceso de evaluación del programa de mejoras en los proyectos productivos de la UCI, basados en el modelo CMMI.* 2010.
31. *Entorno virtual de aprendizaje. Ingeniería de software 2. Clase práctica para estudiantes . UCI .* 2010-2011.

BIBLIOGRAFÍA

1. [En línea] http://calisoft.uci.cu/index.php?view=article&catid=10:art&id=17:expediente-de-proyecto&option=com_content&Itemid=24.
2. [En línea] www.alegsa.com.ar/aplicacion.php..
3. [En línea] www.alegsa.com.ar/procesador%20de%20textos.php.
4. [En línea] <http://www.programacion.org>.
5. Janium.com. *Aplicaciones basadas en Web*. [En línea] 2009. [En línea] 2010. <http://www.janium.com/page2/page1/page6/page7/page7.html>..
6. Masternewmedia.org. *Beneficios De Las Aplicaciones Basadas En Web Y El Anuncio De Microsoft De La Era “En Vivo”*. [En línea] 2005. Be http://www.masternewmedia.org/es/aplicaciones_web/temas_de_aplicaciones_web/Beneficios_De_Las_Aplicaciones.
7. [En línea] <http://www.Masternewmedia.org>.
8. [En línea] <http://www.tecnologiadiaria.com/2009/11/crear-documentos-pdf-acrobat-buzzword.html>..
9. [En línea] <http://www.desarrolloweb.com/articulos/1681.php>..
10. [En línea] <http://decsai.ugr.es/~cb/CSharp/lenguaje/intro.xml>.
11. [En línea] 16- Microsoft Corporation. (2010). Recuperado el 14 de 02 de 2010, de MSDN Visual Studio: <http://msdn.microsoft.com/es-es/library/aa288436%28VS.71%29.aspx>.
12. Subgurim.NET. [En línea] <http://www.subgurim.net/Articulos/asp-net-general/3/por-que-asp-net.aspx>..
13. [En línea] <http://www.info-ab.uclm.es/asignaturas/42580/introdotnet.pdf>
14. [En línea] <http://social.msdn.microsoft.com/Search/es-es?query=Remoting>
15. [En línea] <http://social.msdn.microsoft.com>

16. [En línea] <http://msdn.microsoft.com/es-es/library/bb822049%28v=vs.90%29.aspx>.
17. [En línea] <http://msdn.microsoft.com/en-us/vstudio>.
18. Guía de Usuario de Enterprise Architect. [En línea] 05 de Enero de 2011. <http://www.sparxsystems.com.ar>.
19. **Enterprise Architect - Herramienta de diseño UML.** Sparxsystems.com . [En línea] <http://www.sparxsystems.com.ar/products/ea.html>..
20. [En línea] <http://ingenierioscomputacionipla.wordpress.com/2008/09/12/enterprise-architect-71-modelado-uml/>..
21. [En línea] Microsoft Corporation. <http://support.microsoft.com/kb/930858/es>.
22. **Pressman, Roger.***Un Enfoque Práctico.*
23. [En línea] <http://msdn.microsoft.com/es-es/library/fa1h9d0d%28v=vs.80%29.aspx>..
24. [En línea] Modelo de Dominio. [Online] [Citado: Febrero 01,2010] Disponible en http://iie.fing.edu.uy/ense/assign/desasoft/practico/hoja8/ejemplos_clase2.pdf.
25. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.***El Proceso Unificado de Desarrollo de Software.* La Habana : Felix Varela, 2004.
26. [En línea] Requerimientos. [Online] [Citado: Febrero 01,2010] Disponible en http://iie.fing.edu.uy/ense/assign/desasoft/practico/hoja8/ejemplos_clase2.pdf.
27. [En línea] http://www.procuraduria.gov.co/descargas/contratacion_2006/licitaciones/licitacion112006bid_Anexo4_4Requerimientosfuncionalesynofuncionalesparte_2.pdf.
28. **Sandó, Lesther Delgado Pérez y José Rolando Pérez.***Herramienta para la revisión y seguimiento de errores de la documentación generada en los Proyectos de la Facultad 7.*
29. **Froufe, Agustín. 2005. ARQUITECTURA Modelo/Vista/Controlador.** [En línea] 2005. http://www.ulpgc.es/otros/tutoriales/java/Apendice/arq_mvc.html..

30. **Autores: Amarilis Peña y Yudisleydis Valencia.** *Trabajo de Diploma Herramienta para automatizar el proceso de evaluación del programa de mejoras en los proyectos productivos de la UCI, basados en el modelo CMMI.* 2010.
31. *Entorno virtual de aprendizaje. Ingeniería de software 2. Clase práctica para estudiantes . UCI .* 2010-2011.
32. **Sandó, Lesther Delgado Pérez José Rolando Pérez.** *Herramienta para la revisión y seguimiento de errores de la documentación generada en los Proyectos de la Facultad 7.*
33. —. *Herramienta para la revisión y seguimiento de errores de la documentación generada en los Proyectos de la Facultad 7.*

ANEXOS

Anexo 1: Diagramas de Clases del Diseño

A.1.1 Diagrama de clase del diseño CU- Mostrar Vista Previa

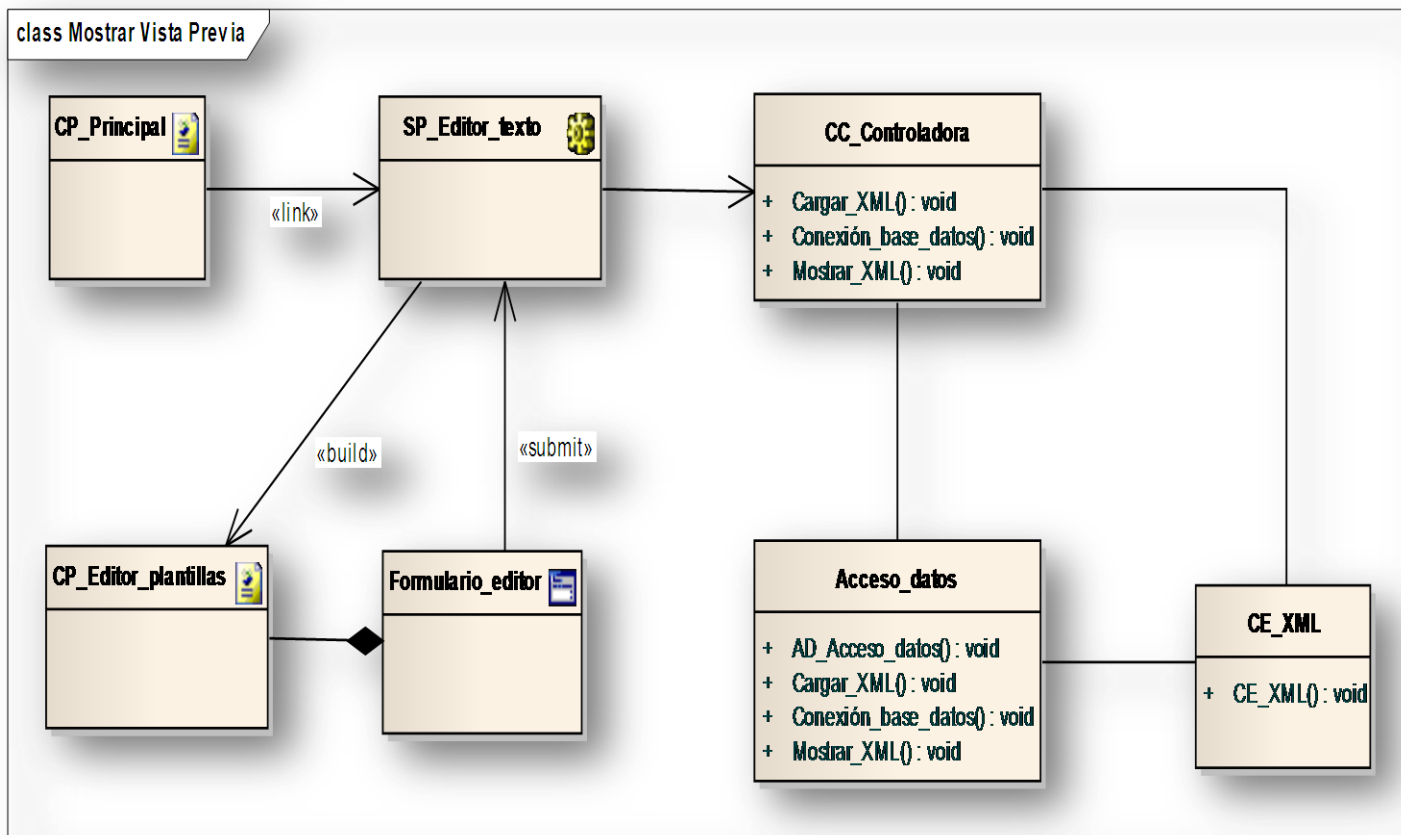


Ilustración 12: Diagrama de clase del diseño CU- Mostrar Vista Previa

A.1.2 Diagrama de clase del diseño CU- Exportar

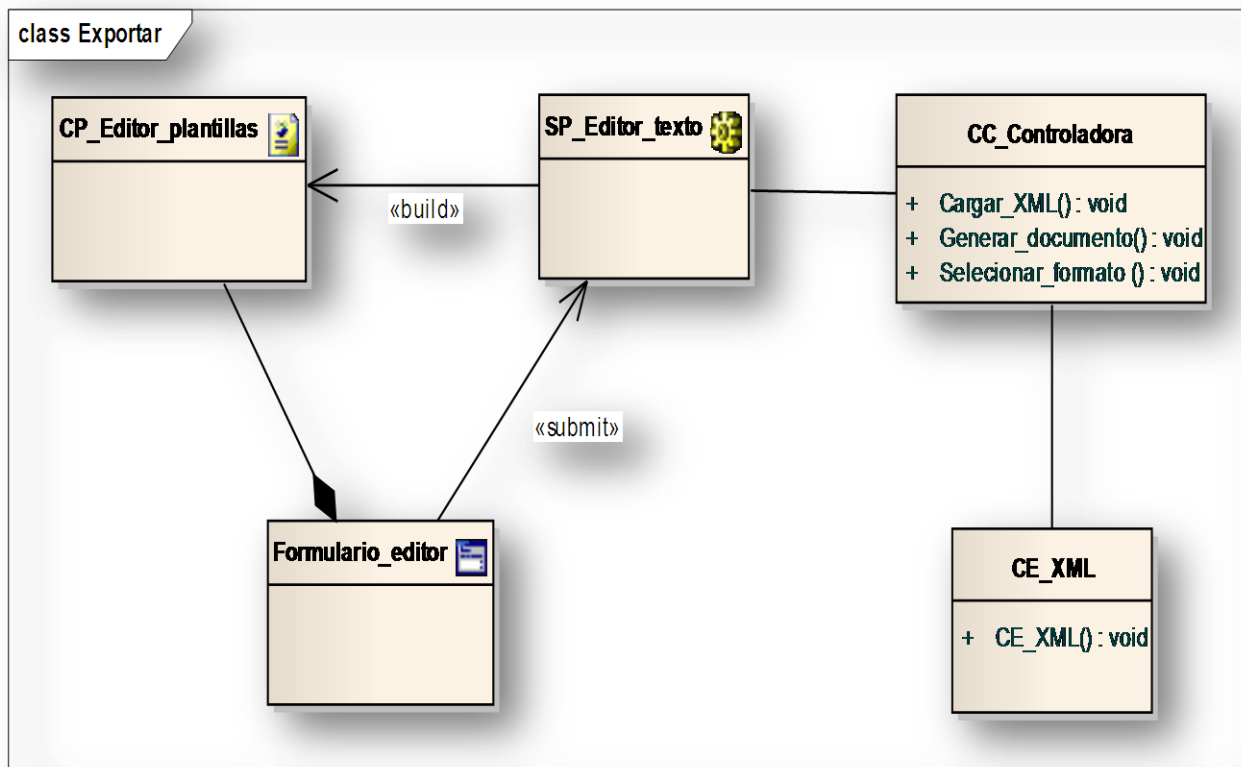


Ilustración 13: Diagrama de clase del diseño CU- Exportar documentos

A.1.3 Diagrama de clase del diseño CU- Guardar Configuración

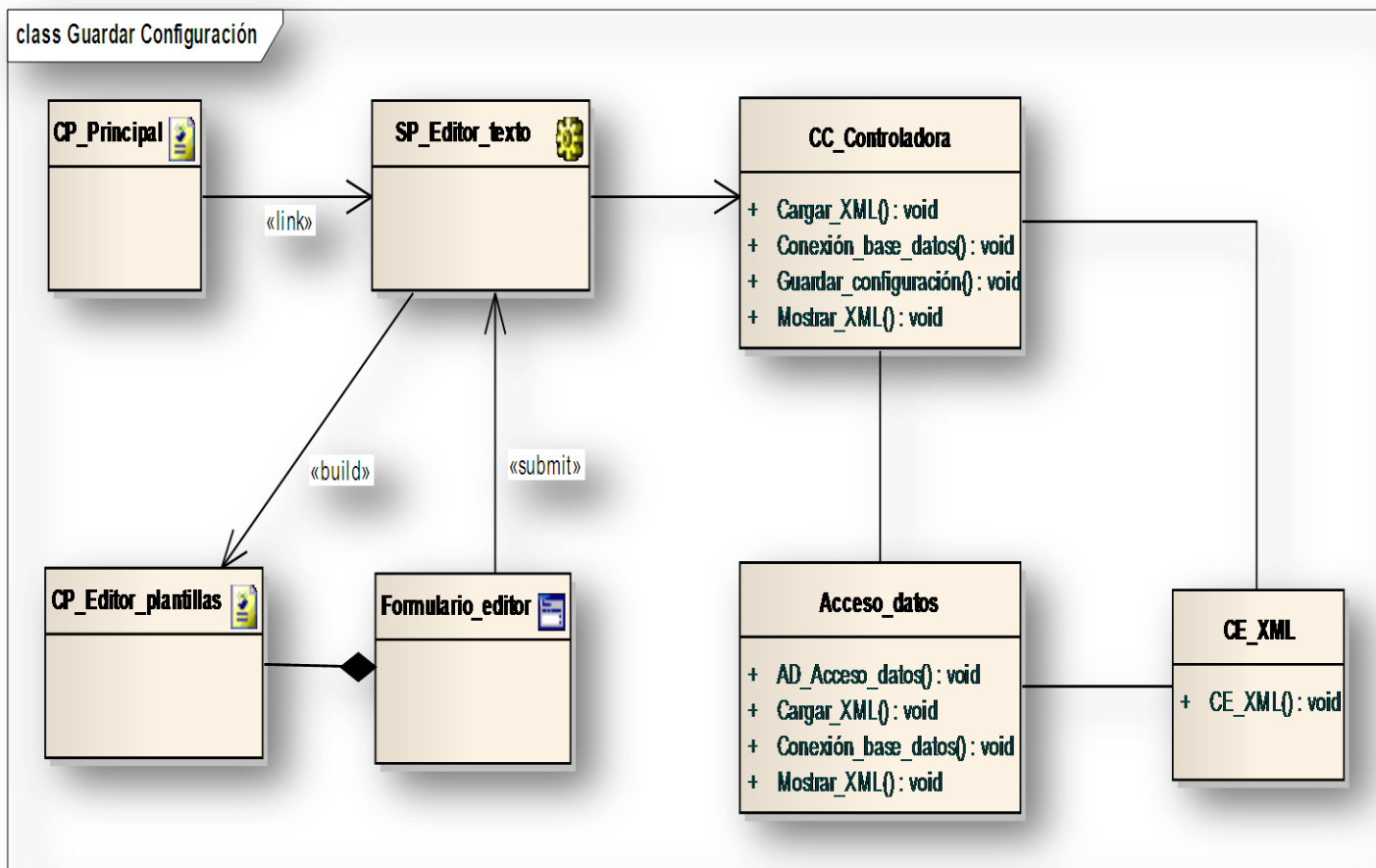


Ilustración 14: Diagrama de clase del diseño CU- Guardar Configuración

Anexo 2: Diagramas de Interacción (Secuencia)

A.2.1 Diagrama de Secuencia del CU- Mostrar Vista Previa

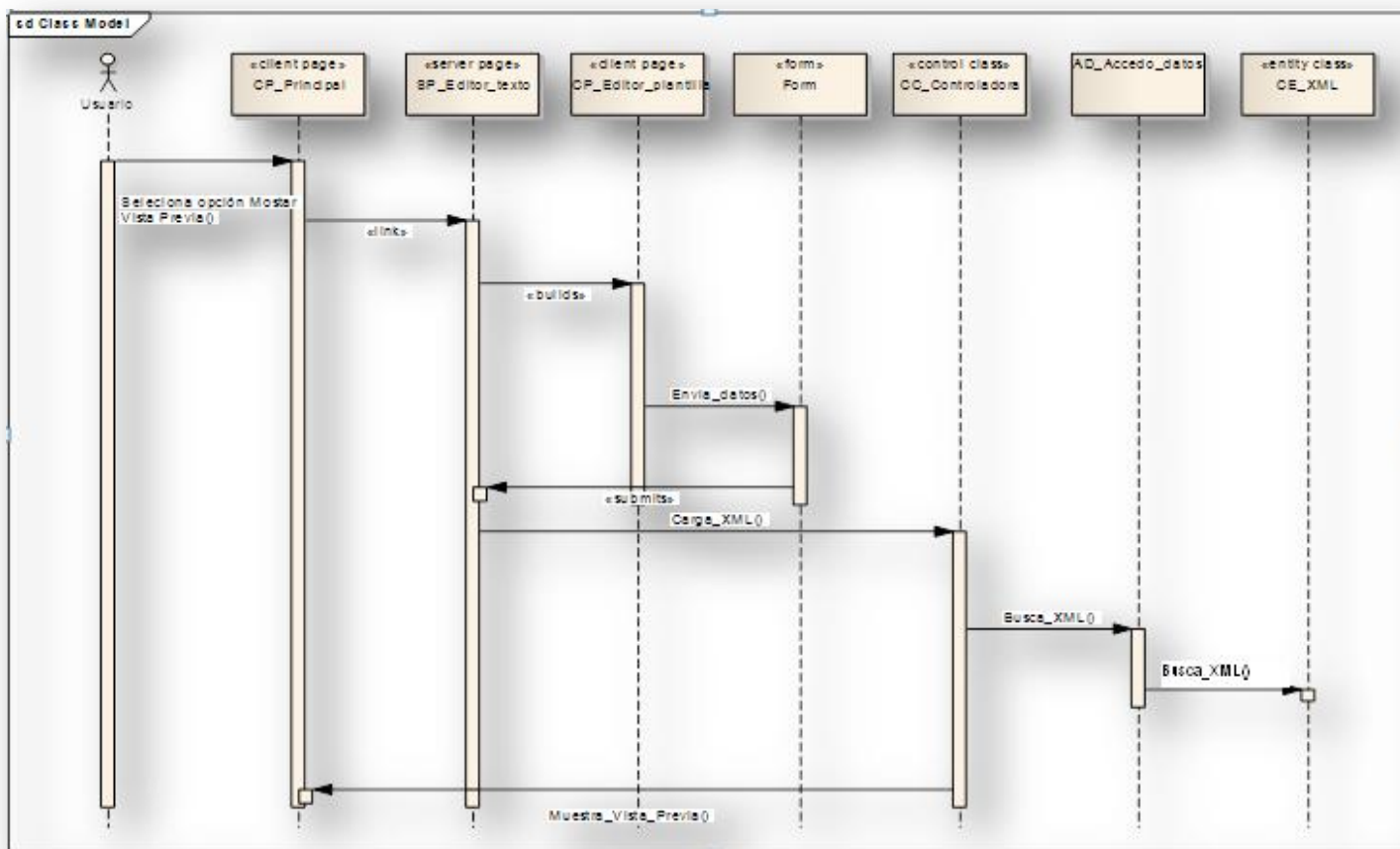


Ilustración 15: Diagrama de Secuencia del CU- Mostrar Vista Previa

A.2.2 Diagrama de Secuencia del CU- Exportar documentos

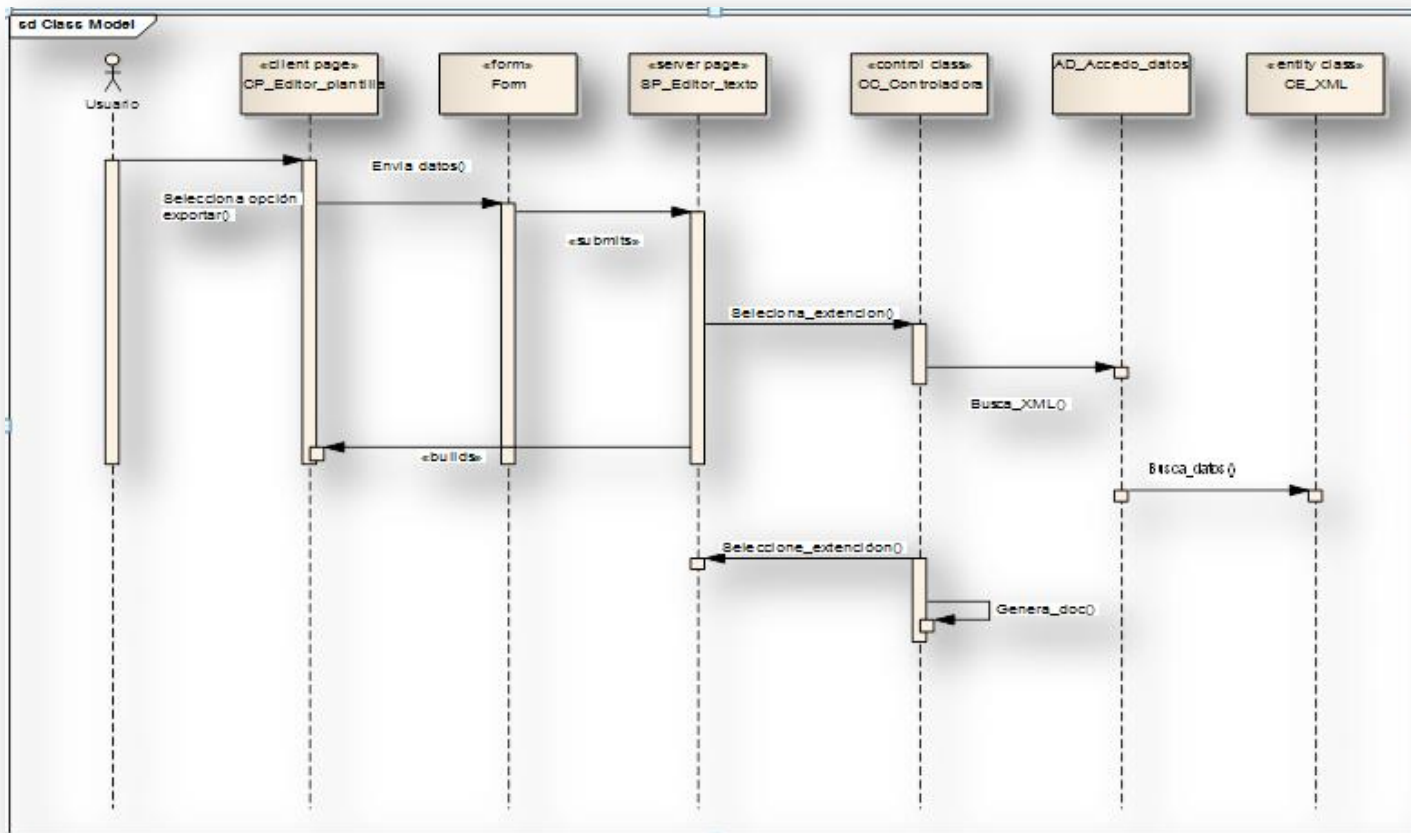


Ilustración 16: Diagrama de Secuencia del CU- Exportar documentos

A.2.3 Diagrama de Secuencia del CU- Guardar Configuración

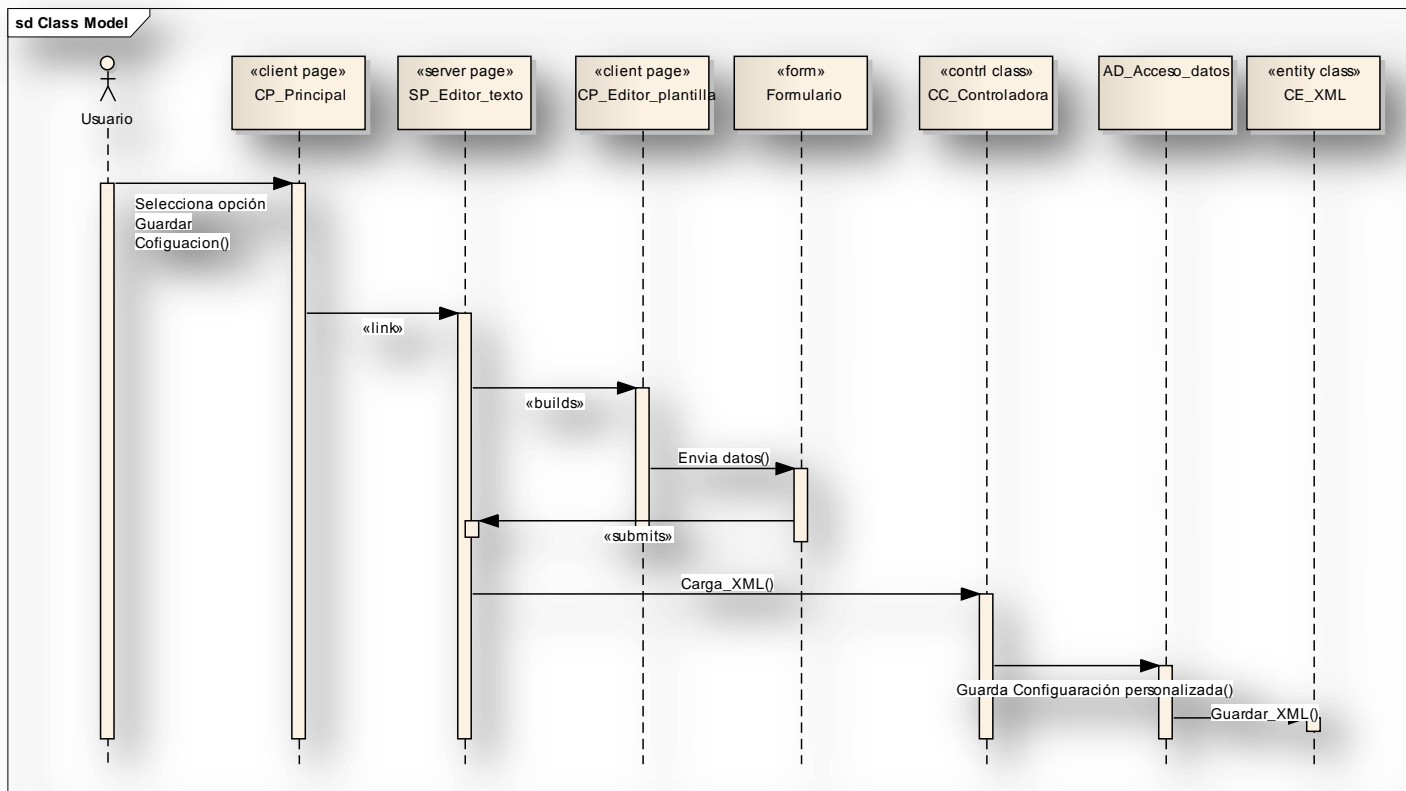
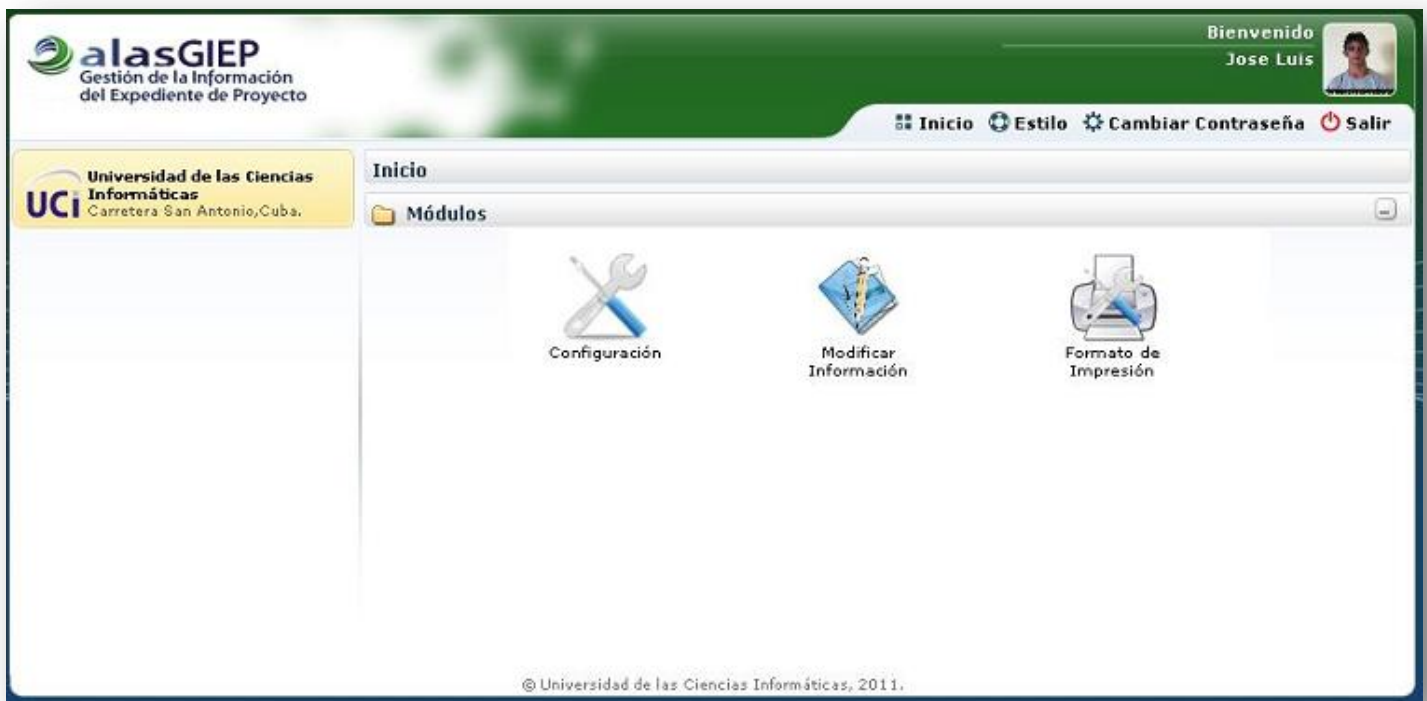


Ilustración 17: Diagrama de Secuencia del CU- Guardar Configuración

Anexo 3: Pantallas del sistema

A.3.1 Interfaz Inicio



A.3.2 Interfaz Mostrar Vista Previa



A.3.3 Interfaz Editar Vista Previa

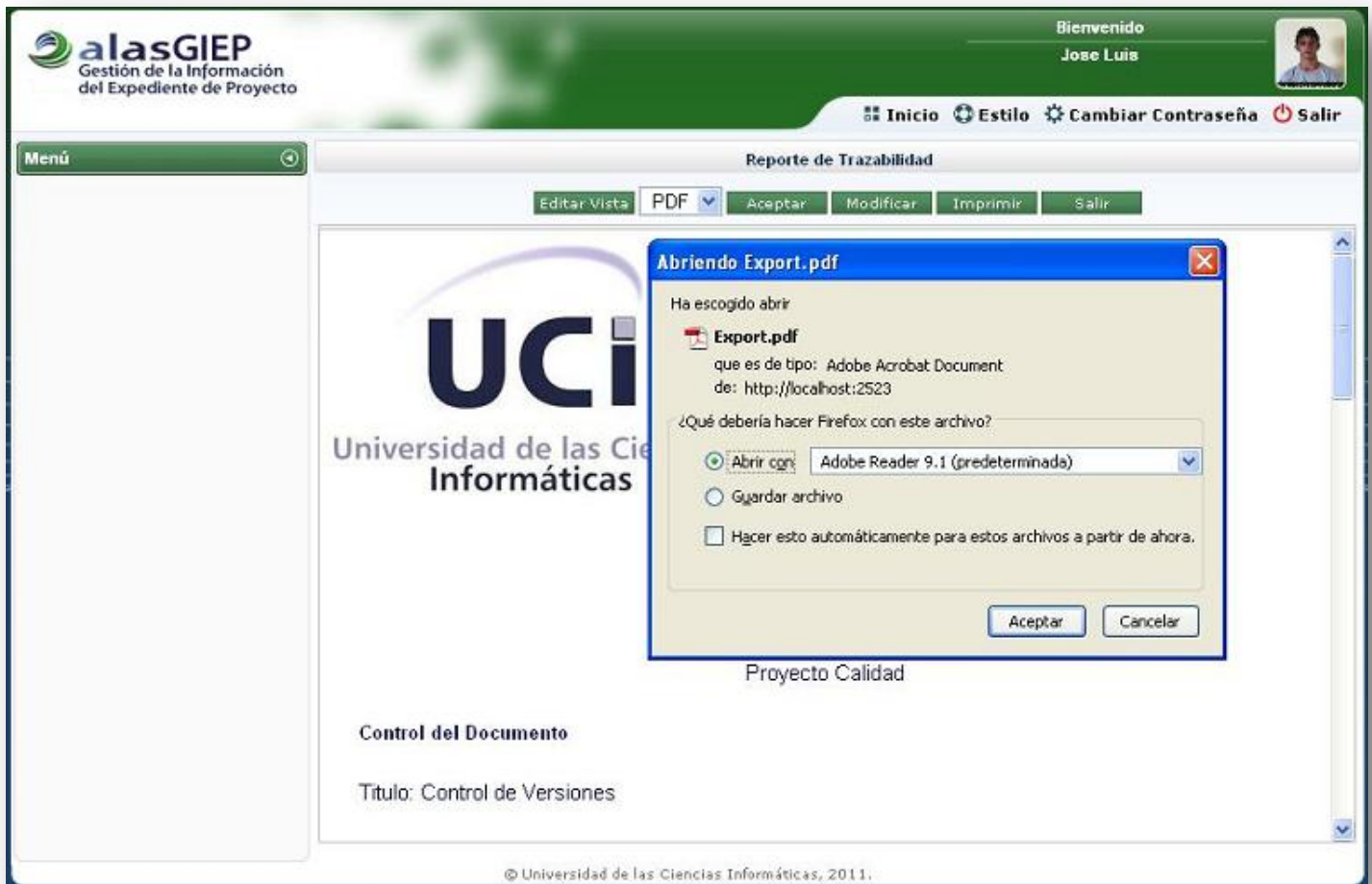
Forma de Distribución: Formato PDF

Control de Cambios

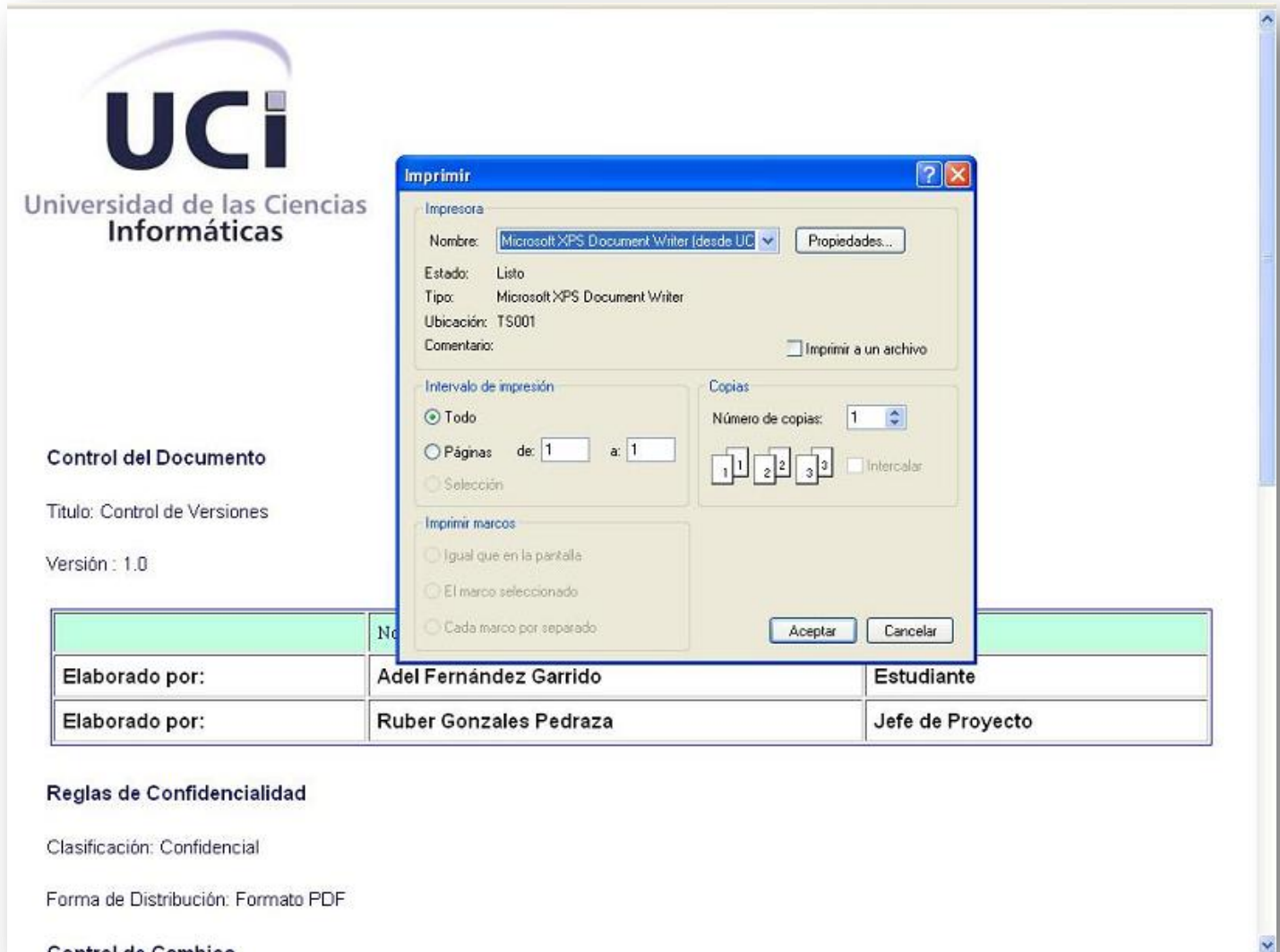
Versión	Sección, Figura, Tabla	Tipo A,B,M	Fecha	Autor del cambio	Descripción del Cambio
1.0	A,22,Elemento	B	10/10/2010	Adel Fernández Garrido	Pobre funcionamiento
2.0	B,2,Elemento1	A	09/10/2009	Damian Almeida	Sin explicaciones
3.0	D,22,ElementoD	M	16/05/2011	Oswaldo Ramirez	No entrego documentación

© Universidad de las Ciencias Informáticas, 2011.

A.3.4 Interfaz Exportar Documentos



A.3.5 Interfaz Imprimir Documentos



GLOSARIO DE TÉRMINOS

.NET: Es una plataforma de Microsoft que provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente en la plataforma.

API: Una interfaz de programación de aplicaciones o API es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

CASE: Sigla que corresponde a las iniciales de Computer Aided Software Engineering; cuya traducción significa Ingeniería de Software Asistida por Computación. Se define como la aplicación de las tecnologías informáticas para la automatización de procesos.

Clases: Conjunto de objetos que comparten atributos, operaciones, relaciones y semántica, las mismas representan los conceptos fundamentales del sistema.

Diagrama de Clases: Representación de los conceptos de importancia en el área de la aplicación, así como de las relaciones entre estos.

Extensible Markup Language (XML): Un lenguaje de marcado extensible que puede usarse para almacenar datos en un formato estructurado, basado en texto y definido por el usuario.

ECMA-334: Estándar internacional que especifica la representación y semántica de programas escritos en C# así como la sintaxis y restricciones de este lenguaje.

Ecma International: Organización internacional que se basa en membrecías de estándares para la comunicación y la información.

HTML (Hyper Text Markup Language): Es el lenguaje para la representación de la información en la web.

HTTP: HyperText Transfer Protocol (Protocolo de transferencia de hipertexto). Es el protocolo usado para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la World Wide Web.

IDE: Siglas en inglés de Integrated Development Environment, en español: Entorno de Desarrollo de Software. Es un programa compuesto por un conjunto de herramientas para un programador. Los mismos

pueden soportar varios lenguajes de programación. Su consistencia está formada por un editor de código, un compilador, un depurador, y un constructor gráfico.

Librería: Conjuntos de código ya realizado que se puede reutilizar en los programas y que ahorran mucho esfuerzo en la programación.

Metodología de desarrollo: Versión amplia y detallada de un ciclo de vida completo de desarrollo de software que incluye reglas, procedimientos, métodos, herramientas, funciones individuales y en grupo por cada tarea, productos resultantes y normas de calidad.

PDF (Portable Document Format): Formato portátil de un documento, formato de archivo que conserva el formateo de documentos y facilita a los usuarios ver un archivo en su forma original desde cualquier computadora a través del programa Acrobat Reader (desarrollado por Adobe Systems).

TCP/IP: El nombre TCP/IP proviene de dos protocolos importantes de la familia, el Transmission Control Protocol (TCP) y el Internet Protocol (IP). En español es Protocolo de Control de Transmisión y Protocolo de Internet. Forma de comunicación básica que usa el Internet, la cual hace posible que cualquier tipo de información (mensajes, gráficos o audio) viaje en forma de paquetes sin que estos se pierdan y siguiendo cualquier ruta posible.

RUP: Sigla anglosajona de Rational Unified Process es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El mismo es un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

WEB: La World Wide Web, la Web o WWW, es un sistema de navegador web para extraer elementos de información llamados "documentos" o "páginas web". Puede referirse a "una web" como una página, sitio o conjunto de sitios que proveen información por los medios descritos, o a "la Web", que es la enorme e interconectada red disponible prácticamente en todos los sitios de Internet.

XML: Lenguaje de marcas utilizado para dar formato a documentos o información en general.