UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 3

Análisis y Diseño de un sistema para la gestión de los procesos de la Dirección de Comunicación Visual

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

Autor: Lázaro Enrique Calixto Varela

Tutor: Lic. Arismayda Dorado Risco

Co-Tutores: D.I. Evelio Gómez Dorado

D.I. Jorge Luis Ortiz Abstengo

Consultante: Ing. Ariel Eduardo Morales Malpica

Ciudad de la Habana, junio 2011

DECLARACIÓN DE AUTORÍA

Declaro ser el único autor de este trabajo y autorizo a la Universidad de Ciencias Informáticas hacer uso del mismo en su beneficio.

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

Lázaro Enrique Calixto Varela

_____

(Autor)

Lic. Arismayda Dorado Risco

_____

(Tutora)

Resumen

La Universidad de las Ciencias Informáticas (UCI) es un centro dedicado a la docencia, producción e investigación creado con el objetivo de contribuir al desarrollo de la informática en Cuba. En ella se combinan estos tres elementos para formar una fuerza especializada en la rama de la informática, capaz de insertar a Cuba en el mercado nacional y mundial, obteniendo productos muy competitivos, de alta calidad estructural y funcional.

La UCI dispone de la Dirección de Comunicación Visual (DCV) que es la encargada de realizar los diseños que son solicitados tanto por los Centros de Desarrollo de Software de la universidad como por entidades externa a la UCI, tramitados por la empresa Albet Ingeniería y Sistemas S.A. (ALBET) directamente, que es la empresa comercializadora.

Esta dirección requiere de un sistema que sea capaz de dar solución a las dificultades administrativas, de control y de divulgación que presenta.

Por lo que el presente trabajo realiza en análisis y diseño del Sistema para la Gestión de los Procesos de la Dirección de Comunicación Visual, con el objetivo de obtener un producto bien especificado. Para ello se identificaron, analizaron y especificaron los requerimientos y casos de uso del sistema. Se generan además los artefactos correspondientes al modelo de análisis y diseño para dar cumplimiento al objetivo previsto. Finalmente se validan los artefactos obtenidos, tanto en el modelo del sistema como en el modelo de diseño en función de asegurar la calidad de los mismos.

Índice

Introducción

Los proyectos han sido desde tiempos prehistóricos una forma de hacer las cosas, aunque en ese entonces no fueran tales ni se nombraran de este modo. La gestión de proyectos es una disciplina que ha ido modificándose a través del tiempo, a la par del avance de las tecnologías y la evolución de los modelos de producción. De esta manera la concepción de planificación y ejecución en base a proyectos es una forma de hacer las cosas que ha venido creciendo y está muy ligado al éxito de las empresas hoy día.

El valor de una buena gestión de proyectos radica, esencialmente; en contar con procesos bien definidos y herramientas adecuadas que faciliten la toma de decisiones en el manejo de los recursos que se involucran en el proyecto, de forma tal que se alcancen resultados predecibles.

La gestión de proyectos es una práctica que las empresas alrededor de todo el mundo han estado adoptando para llevar a cabo todo lo que esté relacionado con sus actividades productivas. Se ha demostrado que ejecutar en base a proyectos ha representado en gran medida la reducción de costos y el aumento considerable de la calidad de los productos y servicios que se ofrecen. (Gido, y otros, 1999)

Cuba es un país que a pesar de estar bloqueado económicamente se ha insertado de lleno en el mundo de las tecnologías, apostando por un futuro de hombres de ciencia. El desarrollo de proyectos de software en nuestro país ha sido potenciado por la dirección de la revolución, pues es una fuente importante de ingresos de divisas al país. La adecuada gestión de estos proyectos es clave para su éxito, pues de ello depende en gran medida el cumplimiento de los acuerdos pactados con los clientes y por consiguiente el prestigio de Cuba en esta rama.

La Universidad de las Ciencias Informáticas (UCI) es un Centro Docente Productor creado con el objetivo de contribuir al desarrollo de la informática en Cuba. En ella se combinan la docencia, la investigación y la producción para formar una fuerza especializada en la rama de la informática, capaz de insertar a Cuba en el mercado nacional y mundial, obteniendo productos muy competitivos, de alta calidad estructural y funcional.

La UCI dispone de la Dirección de Comunicación Visual (DCV) que es la encargada de realizar los diseños que son solicitados tanto por los Centros de Desarrollo de Software de la Universidad como por entidades externas a la UCI, tramitados por la empresa Albet Ingeniería y Sistemas S.A. (ALBET) directamente, que es la empresa comercializadora. Es menester de la DCV proporcionar un diseño que se ajuste a las características de la solicitud hecha, en el menor plazo de tiempo posible y con la calidad requerida.

INTRODUCCIÓN

Hoy día la labor que se realiza en la DCV es poco conocida dentro de la universidad, pues no cuenta con un portal que promocione su quehacer cotidiano, los servicios que ofrece, ni la forma de acceder a éstos. Esta situación trae consigo de igual forma, una pérdida de oportunidades tanto a los neófitos, como a los más experimentados, por no conocer en profundidad toda la ayuda en materia de Comunicación Visual con que pudieran contar para sus proyectos.

Actualmente el proceso de solicitud a la DCV se realiza de forma manual mediante una planilla que luego de ser llenada por el jefe de proyecto, pasa a la dirección del centro de desarrollo de software correspondiente. Este se encarga de entregarla a la DCV, que es la entidad que decide si acepta o no la petición realizada. Este procedimiento apareja una pérdida de tiempo considerable, que en muchas ocasiones repercute en el plazo que se dispone para dar respuesta a la solicitud. Además, los clientes de la DCV no conocen el estado en que se encuentra la solicitud que realizan a esa entidad, lo que hace necesario que se cree un mecanismo que minimice el tiempo destinado a los trámites de realización de solicitud y que le permita a los clientes tener datos reales y precisos del estado actual de sus solicitudes.

Otro importante contratiempo presente en la DCV es el control y seguimiento de las actividades que allí se realizan, puesto que los ejecutivos que lo guían forman parte de los equipos de trabajo, lo cual hace engorroso la realización de revisiones periódicas a las tareas de sus subordinados para verificar el estado que presenta. No tienen a su disposición una forma rápida y eficiente de conocer los proyectos en los que están inmersos los trabajadores, dificultándose así la toma de decisiones para la asignación de trabajos. Todos estos problemas conllevan a la necesidad de crear un sistema que minimice el tiempo necesario para llevar un adecuado control de las actividades que realizan los diseñadores, además de facilitar la planificación interna de sus integrantes.

Se hace necesario contar además con una forma ágil de evaluar el desempeño de los trabajadores de la DCV, pues actualmente este proceso se realiza de forma semi-automática, lo cual acarrea pérdida importante de tiempo y recursos materiales.

De esta forma se propone como **problema a resolver**: ¿Cómo transformar las necesidades de los trabajadores de la DCV a un lenguaje entendible por los desarrolladores?

Como **objetivo general** se tiene: Realizar el análisis y el diseño de un sistema para la gestión de los procesos de la Dirección de Comunicación Visual.

El **objeto de estudio** identificado es: la gestión de proyectos, centrando el **campo de acción** en los Sistemas de Gestión de Proyectos para los procesos de la DCV de la UCI.

La **idea a defender** es: Desarrollando el análisis y diseño del sistema para la gestión de los procesos de la Dirección de Comunicación Visual, se contribuirá a una posterior implementación del mismo.

Además se proponen los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación.
- Realizar el análisis y el diseño del sistema.
- Validar la solución propuesta.

Para el cumplimiento de los objetivos específicos se plantean una serie de **tareas**, dentro de las cuales se encuentran:

- Analizar las herramientas más usadas en el mundo y en la UCI sobre gestión de proyectos.
- Analizar las principales herramientas y lenguajes de modelado más usados internacionalmente.
- Analizar los elementos relacionados con la Ingeniería de Requisitos.
- Analizar los elementos relacionados con el diseño.
- Caracterizar los métodos de validación.
- Modelar y describir los procesos.
- Identificar requisitos funcionales y no funcionales.
- Elaborar el documento Especificación de Requisitos.
- Elaborar el documento Modelo de Sistema.
- Realizar los diagramas de secuencia, clases del diseño y modelo de datos.
- Aplicar métricas para evaluar la calidad de los artefactos generados.

Para esta investigación se utilizaron los siguientes métodos de la Investigación Científica:

Métodos Teóricos

Analítico –Sintético: Para entender y resumir la situación que presentan los clientes, además para interiorizar y sintetizar el estudio que se realiza a lo largo del capítulo uno, incluyendo estudio del estado del arte sobre las herramientas más usadas en la gestión de proyectos, herramientas a utilizar para el desarrollo del sistema y los principales conceptos relacionados con la investigación.

Modelación: Para la creación de modelos y diagramas que reflejen la lógica del Sistema para la Gestión de los procesos de la DCV.

Métodos Empíricos

Entrevista: Para interactuar con los clientes e identificar los requerimientos del sistema.

Observación: Para observar detenidamente los procesos de la DCV, de esta forma lograr un entendimiento profundo del funcionamiento de esta entidad.

Resumen por capítulo

Capítulo 1. Fundamentación teórica y tecnologías actuales

El capítulo 1 expone un estudio de todos los temas concernientes a la investigación. Se analizan las herramientas más usadas en el mundo y en la UCI para la gestión de proyecto, las herramientas y lenguajes de modelado. Se repasan además las metodologías más empleadas en el desarrollo de software y términos relacionados con el análisis y diseño.

Capítulo 2. Modelado del negocio, levantamiento de requisitos y diseño del sistema.

El capítulo 2 expone la solución que se propone para el cumplimiento del objetivo general. En él se modelan y se describen los procesos de la DCV, se realiza el levantamiento de requisitos del sistema, que incluye la descripción de los requisitos y de los casos de usos en los que fueron agrupados los requisitos funcionales y se obtiene la descripción detallada del sistema en cuestiones de diseño.

Capítulo 3. Validación de la solución propuesta.

En el capítulo 3 de realiza un análisis de los resultados obtenidos en este trabajo. Se muestran los procedimientos y métodos encaminados a medir la factibilidad de los artefactos obtenidos tanto en el modelado del sistema como en el diseño. Se hace uso de método de Kano para medir el grado de satisfacción del cliente con respecto a los requerimientos elicitados, así como el uso de métricas para validar los artefactos del modelado del diseño.

CAPÍTULO 1
FUNDAMENTACIÓN TEÓRICA Y TECNOLOGÍAS ACTUALES

Capítulo 1: Fundamentación teórica y tecnologías actuales

1.1 Introducción

El presente capítulo tiene como objetivo abordar los aspectos teóricos fundamentales que facilitan la comprensión y el desarrollo de la solución que se desea proponer. Se realiza un estudio de las principales herramientas y aplicaciones que existen para la gestión de proyectos, las cuales son utilizadas tanto en el mundo como en la Universidad de las Ciencias Informática. Se brinda un resumen de las herramientas a utilizar en la realización del sistema así como la metodología RUP (*Rational Unified Process* por sus siglas en inglés) (Proceso Unificado de Software) y los lenguajes de modelados más referenciados en el mundo.

1.2 Herramientas más usadas para la gestión de proyectos

En la gestión de proyectos la asignación de tareas y su seguimiento son parte importante que influyen en el desempeño de los miembros del equipo de trabajo. Su función principal es indicarle a cada trabajador las tareas a realizar en un tiempo determinado. Se le debe dar continuidad para verificar el avance de la tarea en cuanto a alcance, tiempo dedicado, recursos, calidad, coste, riesgos etc. (Gido, y otros, 1999)

Las herramientas de gestión de proyectos en general proporcionan un trabajo colaborativo con los diferentes miembros de un proyecto, permiten realizar un seguimiento de las tareas, realizar la programación, gestionar los recursos asignados y gestionar los documentos de estos.

Las herramientas que se analizan a continuación poseen un grupo importante de características universales e imprescindibles para un sistema de gestión de proyectos. Estas características, como es lógico, se incluyen en el sistema que se modela. No se seleccionan las herramientas que se mencionan más adelante, debido a que los procesos de la DCV no son soportados por éstas.

Los trabajadores de esa dirección requieren un sistema dinámico que interactúe en tiempo real con todos los trabajadores y sus respectivos proyectos, propiciándole en todo momento una vista panorámica del estado en que se encuentran los proyectos en que se encuentran trabajando. Se requiere además poder visualizar una especie de diagrama de Gant, con particularidades especificadas por los clientes; las herramientas analizadas más adelante no brindan esta funcionalidad.

A continuación se analizan dos herramientas que dan soporte a la gestión de proyectos:

1.2.1 Microsoft Project

Herramienta de gestión de proyectos que fue realizada para el Sistema Operativo Windows. Se emplea en la planeación y el control de los proyectos. Es eficiente para planear todo tipo de trabajos, desde proyectos sencillos hasta grandes programas. Presenta varias funcionalidades entre las que se encuentra el calendario que es visualizado mediante un Diagrama de Gantt, que es una herramienta grafica para modelar la duración de un proyecto en término de tareas y actividades, permitiéndole al usuario representar la ruta crítica del proyecto. Al introducir las tareas permite asignar responsables y duración de la misma. También permite llevar el control del por ciento de desarrollo a medidas que se van ejecutando las actividades. Le facilita al usuario llevar la trazabilidad por tareas, designar tareas dentro de otras y resaltar los hitos de cada planificación.

El proyecto ofrece una mejor coordinación entre los equipos a través de sus notificaciones automatizadas, integración con otros programas y acceso a portal web, que permite a los usuarios ver fácilmente, actualizar y analizar la información del proyecto. Es una herramienta privativa. (Microsoft, 2007)

1.2.2 Redmine

Herramienta web de gestión de proyectos muy flexible que brinda soporte a múltiples proyectos simultáneamente, permitiendo realizar notificaciones por correo electrónico de las diferentes tareas o peticiones orientadas a los miembros del proyecto. Esta herramienta permite publicar actividades de ámbito general que son de conveniencia para los miembros del proyecto (Philippe Lang, 2006-2011). Da la posibilidad de hacer la planificación por etapas, facilitando la organización del equipo de desarrollo, además es preciso crear iteraciones que son creadas según las características del proyecto.

El Redmine facilita la realización de diagramas de Gantt y calendarios de tareas así como seguimiento de estas, además permite inserción de noticias, documentos y archivos para el conocimiento de todos los que integran un grupo de trabajo. (Philippe Lang, 2006-2011)

Presenta compatibilidad con varios idiomas, integración con varias bases de datos que sirven de apoyo, y permite la integración con subversion como SVN, CVS, Mercurial y Bazaar.

Una vez creados los usuarios se pueden crear grupos de usuarios que son útiles para tener adecuadamente recogidos o registrados los grupos de trabajo y los equipos de desarrollo. También se definen los perfiles de los roles de usuarios indicando claramente todos los roles que intervienen en el proyecto en general.

Otra funcionalidad disponible de esta herramienta es que permite llevar un correcto control y seguimiento de los riesgos del proyecto y facilita la comunicación mediante foros. Luego de creado éste, será posible recibir y enviar mensajes, editarlos, modificarlos o borrarlos. Permite la creación de wiki por proyectos y genera información en diagramas de Gantt. Soporta autenticación LDAP. (Philippe Lang, 2006-2011)

Es una herramienta de código abierto, multiplataforma y en la UCI es una de las más utilizadas en la gestión de proyectos.

1.3 Metodologías de desarrollo

Las metodologías de desarrollo de software pueden clasificarse como un conjunto de pasos lógicos y ordenados ya preestablecidos que deben seguirse para desarrollar software. Estas metodologías persiguen un fin común, que es convertir el desarrollo de software en un proceso formal, con resultados predecibles que permitan obtener un producto con la calidad requerida en un tiempo razonable. (Sommerville, 2005)

La elección de la metodología a utilizar en el desarrollo de software está supeditada a las características del producto que se desea desarrollar así como el equipo de desarrollo que lo va a ejecutar. Para la elección de una u otra metodología es importante que esta sea de fácil asimilación por los integrantes del equipo y fácil de aplicar, además de aportar beneficios tangibles al proyecto.

1.3.1 Extreme Programing (XP)

Extreme Programing (programación rápida o extrema) es una metodología que se emplea en proyectos cortos, de poca complejidad y de poco personal. Es además una de las metodologías de desarrollo más usadas actualmente.

Dentro de sus principales características incluye las **pruebas unitarias** que son las pruebas que se realizadas a los principales procesos para determinar posibles fallas que pudieran ocurrir. Es como si se estuviera adelantando a obtener los posibles errores.

Otra de sus características se encuentra la **re-fabricación**, que consiste en la reutilización de código existente de modo que se facilite el trabajo a los integrantes del equipo de desarrollo y los cambios que se realicen sean más cómodos de ejecutar.

Un aspecto importante es la **programación en pares**, lo que requiere de gran comunicación principalmente entre cada uno de los dúos que se formen. Su objetivo es lograr un trabajo coordinado entre los dos miembros, de forma que cada miembro conozca a la perfección lo que en conjunto se realiza.

En esta metodología el cliente o el usuario forma parte del equipo de desarrollo. Siempre se parte de lo pequeño o simple, y posteriormente se complementan con las nuevas funcionalidades que se le adicionan a medidas que se requieran de estas mediante un proceso de retroalimentación continua. (Hurtado, y otros, 2005)

El desarrollador tiene el derecho de implementar los procesos como estime conveniente, crear el sistema con la mayor calidad posible, pedirle al cliente en cualquier etapa del desarrollo del software explicación sobre determinado aspecto del negocio que no conozca así como realizar propuestas de determinadas funcionalidades que estime deba ir en el sistema. (Wake, 2002)

1.3.2 RUP

La metodología RUP (Rational Unified Process) es una metodología pesada, que por sus características genera una amplia documentación que facilita el trabajo a los que la practican. RUP es un proceso de desarrollo de software que contribuye a mejorar el rendimiento del equipo de trabajo en cuanto a productividad. Define claramente los roles, responsabilidades y actividades de todos los que intervienen en este proceso. (Jacobson, y otros, 2000)

Presenta tres características fundamentales que la distinguen de las demás:

- Dirigida por casos de usos porque todo lo que se quiere construir en el sistema (requisitos) son agrupados de acuerdo a sus peculiaridades en casos de usos que son los que guían el proceso.
- Centrada en la arquitectura porque no existe un modelo único que abarque todos los elementos del sistema sino que existen múltiples modelos y vistas que definen la arquitectura de software de un sistema.
- Iterativa e incremental porque cuenta con cuatro fases dentro de las que se encuentra la fase de **inicio**, **elaboración**, **construcción** y **transición**, estas a su vez se dividen en un grupo de iteraciones que son nombradas flujos de trabajo que van añadiendo completitud al producto que se quiere obtener.

Los flujos de trabajos son: **modelado del negocio**, **requisitos**, **análisis y diseño**, **implementación**, **prueba**, **despliegue**, **gestión de configuración y cambios**, **gestión de proyectos y ambiente.**

La fase de **inicio** se encarga de obtener un acuerdo entre todos los interesados, o sea, clientes, usuarios y desarrolladores del software. En ella se delimitan las condiciones por la cual se necesita un software, los elementos que van incluidos en el producto final y las que no, se realiza una estimación del coste total, una planificación general de todo el proyecto y se analizan los principales riesgos. El hito de esta fase son los objetivos del sistema. (Kruchten, 2000)

La fase de **elaboración** se encarga de obtener una línea base de la arquitectura lo suficientemente estable para el desarrollo del sistema, se capturan los requisitos y se tratan los riesgos arquitectónicamente significativos.

La fase de **construcción** es la que se encarga de desarrollar el sistema. En ella se obtienen las funcionalidades deseadas que se definieron como requisitos y se trata de conseguir la calidad máxima en cada versión generada en las diferentes iteraciones. Su hito es la capacidad operativa del sistema.

La fase de **transición** es la última que se realiza y en ella se tiene el producto preparado para la entrega al cliente, o sea, el ¨release¨ del producto. También se enseña al usuario a utilizar el software.

El flujo de trabajo de **modelado del negocio** es considerado como la base para entender el funcionamiento de la empresa en aras de comprender la interacción entre los procesos que la componen.

El flujo de trabajo de **requisitos** define qué el sistema debe hacer, en él se explica cómo obtener las necesidades de los clientes para ser incluidas en el sistema.

El flujo de **análisis y diseño** describe cómo el sistema debe realizarse, por lo que indica con precisión lo que se debe implementar. (IBM, 2006)

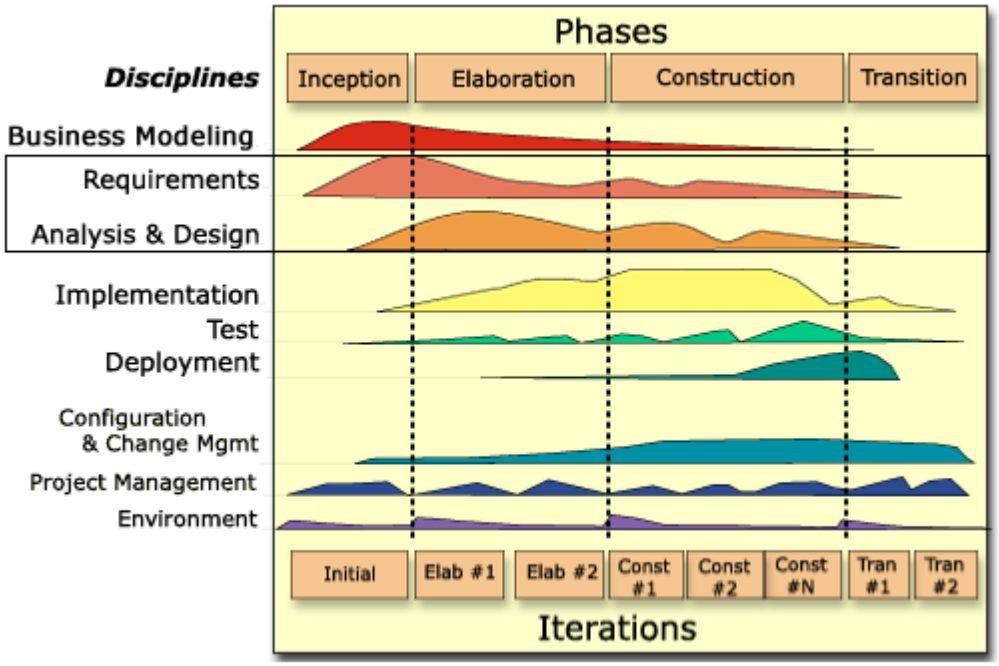


Figura 1 Proceso unificado de Desarrollo de Software (fases y flujos de trabajos)

1.4 Lenguajes de Modelado

Los modelos de procesos del negocio describen de forma entendible cómo funciona el negocio de una empresa. De igual forma describen las actividades involucradas en el negocio y cómo se relacionan unas con otras, así como su interacción con los diferentes recursos que intervienen en el flujo de los procesos para lograr una meta u objetivo.

Los modelos de procesos de negocio (MPN) poseen un uso de amplio espectro tales como el desarrollo de sistemas para la automatización de procesos, la simulación así como la re-ingeniería de procesos. Los MPN pueden ser hechos usando varios lenguajes para este fin, los cuales son diferentes entre sí, sobre todo porque cada uno tiene una forma diferente de interpretar los procesos, esto depende en gran medida del propósito de su creación.

Para el modelado de procesos de negocio y modelado de sistemas informáticos existen múltiples lenguajes de modelado, a continuación se explican algunos de ellos:

1.4.1 BPMN

BPMN (*Business Process Modeling Notation*) como su nombre lo indica es una notación para modelar procesos de negocio de una empresa. El objetivo principal de BPMN es proporcionarle a todas las personas involucradas en el negocios una notación factible para entender cómo

funcionan los procesos que componen una empresa, desde el analista de negocio que es el encargado de hacer el borrador inicial, pasando por los desarrolladores que son los encargados de implementar todo el sistema que ejecutará dichos procesos, hasta las personas que son los encargados de ejecutar y gestionar los procesos.

BPMN es considerado como un mecanismo simple para crear modelos de procesos de negocio a través de las cuatro categorías básicas de elementos: objetos de flujo, objetos conectores, rol de proceso, artefactos.

Dentro de los objetos de flujo están los eventos, las actividades y *gateway*. Entre los objetos conectores están los flujos de secuencia, flujos de mensaje y asociaciones. El rol de proceso está representado por las *pool* y las *lane*. Y entre los artefactos están los objetos de datos, los grupos y las anotaciones.

BPMN está basado en las experiencias aportadas por otras notaciones y se encarga de aglutinar las mejores prácticas de algunas de ellas. Se pensó y se diseñó con la intención de crear una notación estándar para modelar proceso. Dentro de las notaciones estudiadas para conformar BPMN están los diagramas de actividades de UML, UML EDOC. (Burcio Sánchez, 2009)

1.4.2 UML

UML (*Unified Modeling Language* o Lenguaje de Modelado Unificado) es un lenguaje de modelado visual y estándar que se emplea para representar, visualizar, construir y documentar los diferentes artefactos que se generan en la construcción de un sistema de software. Se emplea con el objetivo de mantener un estricto control sobre los artefactos y de manejar de forma organizada la información que surge a lo largo del ciclo de vida de este. Brinda un vocabulario y un conjunto de reglas que facilitan la comunicación, que en este caso es mediante una representación gráfica. Está definido para un proceso de desarrollo de software iterativo, como es el caso de RUP y brinda apoyo a los procesos orientados a objetos. (Rumbaugh, y otros, 1998).

Un modelo UML indica qué es lo que supuestamente hará el sistema, no precisamente cómo lo hará, pero constituye una guía y una representación de las especificaciones del sistema, ayudando a tomar decisiones para lograr un sistema que cumpla con todas las expectativas del cliente y los usuarios.

Cuenta con un conjunto de diagramas que brindan organización. Están los diagramas de estructura estática, los de comportamiento y los de implementación.

Diagramas de estructura estática

- Diagramas de casos de uso.
- Diagrama de clases.
- Diagrama de objetos.

Diagramas de comportamiento

- Diagrama de interacción.
- Diagrama de secuencia.
- Diagrama de colaboración.
- Diagrama de actividad.
- Diagrama de estados.

Diagramas de implementación

- Diagrama de componentes.
- Diagrama de despliegue.

1.5 Lenguajes de programación web

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de los usuarios. Inicialmente se dieron soluciones mediante lenguajes estáticos, a medida que pasó el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación web dinámicos, que permitieran interactuar con los usuarios además de lograr mayor productividad y calidad al incorporarlos. Es posible contar con varios lenguajes de programación para desarrollar en la web, tanto del lado del cliente como del servidor.

Los lenguajes del lado del cliente son aquellos capaces de ser dirigidos directamente por el navegador sin necesidad de un pre-tratamiento; por ejemplo:

- **Javascript** es un lenguaje interpretado creado por Brendan Eich en la empresa *Netscape Communication*s. El código Javascript puede ser integrado dentro de las páginas web. El W3C para evitar incompatibilidades diseñó un estándar para la modelación de objetos del documento, más conocido como DOM (en inglés *Document Object Model*). Entre sus

ventajas está su scripting seguro y fiable y contradictoriamente su debilidad es su visibilidad al alcance de cualquier usuario.

- **HTML** (acrónimo en inglés de *Hyper Text Markup Language*, en español Lenguaje de Marcas Hipertextuales), es un lenguaje estático para el desarrollo de sitios web, desarrollado por *World Wide Web Consortium* (W3C). Los archivos pueden tener las extensiones htm, html. Entre sus ventajas se encuentra que es admitido por todos los exploradores, sencillo y de archivos pequeños. Como desventaja resalta que la interpretación de cada navegador puede ser diferente.

También están los lenguajes para desarrollo web del lado del servidor, estos son reconocidos, ejecutados e interpretados por el propio servidor y se envía al cliente en un formato entendible por este.

Entre los lenguajes más comunes del lado del servidor tenemos:

- **ASP.NET** es un lenguaje comercializado por Microsoft. Fue desarrollado para resolver las limitantes de su antecesor ASP. Para el desarrollo de ASP.NET se puede utilizar C#, VB.NET o J#. Creado para desarrollar tanto web sencillas o grandes aplicaciones. Para su funcionamiento se necesita tener instalado IIS (*Internet Information Server*) con el Framework .Net. Entre sus ventajas se encuentra que es un lenguaje completamente orientado a objetos, su velocidad de respuesta del servidor y seguridad entre otras. El consumo de recursos es su desventaja.
- **PHP** es un lenguaje de script interpretado utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Lenguaje multiplataforma, fácil de aprender y no requiere definición de tipos de variables, siendo estas sus mayores ventajas. Entre sus desventajas se encuentra su pobre programación orientada a objetos para aplicaciones grandes, además de dificultar la organización por capas de la aplicación.

La integración entre lenguajes de programación web del lado del servidor y del lado del cliente posibilitan la implementación y con esta la solución a muchos problemas actuales. (World Wide Web Consortium, 1994)

1.6 Herramientas CASE

Las herramientas CASE (*Computer Aided Software Engineering*) son sistemas informáticos que apoyan el proceso de desarrollo de software con el objetivo de aumentar la productividad y reducir el coste en tiempo de realización y dinero a invertir. Dentro de las principales herramientas CASE usadas en la UCI están:

1.6.1 Enterprise Architect

Enterprise Architect (EA) es una herramienta de diseño y análisis UML, que da soporte a las principales etapas de desarrollo de software, desde los requisitos hasta el despliegue. Es considerada una herramienta multiusuario, que funciona sobre plataforma Windows. Facilita la generación de documentación flexible y de muy alta calidad.

Tienes poderosas funcionalidades como conexión a repositorios que facilitan el trabajo en equipo, control de versiones, interfaz, generación de documentos y soporta generación e ingeniería inversa de código fuente para un amplio número de lenguajes, como C++, C#, Java, Delphi, VB.Net, Visual Basic y PHP. (Sparx Systems Pty. Ltd.)

Permite la representación de artefactos mediante diagramas UML, pero también soporta lenguajes de modelados del negocio como BPMN y mapas conceptuales. Brinda seguridad dependiendo de las necesidades del usuario, facilitando la asignación de permisos tanto para usuarios independientes como a grupos de usuarios. Facilita la administración de requisitos ayudando a establecer un modelo jerárquico de requisitos con buena organización.

Provee una generación poderosa de documentos y herramientas de reporte con un editor de plantilla completo.

1.6.2 Visual Paradigm

Visual Paradigm for UML (VP-UML) es una herramienta CASE para UML, que puede ser utilizado por todas las personas interesadas en construir sistemas de software.

Dentro de sus características se encuentra la interfaz de recursos céntricos, que permite el acceso a las opciones de modelado fácilmente, sin necesidad de desplegar el menú o ir a la barra de herramientas.

VP-UML se caracteriza por:

- Permitir la integración con entornos de desarrollo como Eclipse, que facilita la generación

de código y la ingeniería inversa.

- Soportar un gran número de diagramas incluyendo especificaciones como UML, BPMN, SysML, ERD y más.
- Facilitar la generación de reportes a formatos como HTML, PDF, Word y otros.
- Permitir importar diagramas que hayan sido creados por IBM Rational Rose y ERWin. Realiza generar código a partir de los diagramas, soportando lenguajes como Java, C#, VB.NET, PHP, ActionScript, C++, Python.

Es una herramienta con licencia comercial y la UCI puede hacer uso de ella.

1.7 Sistemas de Gestión de Contenido

Los Sistemas de Gestión de Contenidos (en inglés Content Management System, CMS), son programas que permiten crear una estructura de soporte para la creación y administración de contenidos, en este caso, en páginas web. Estas aplicaciones informáticas son usadas para crear, editar, gestionar y publicar contenido digital en diversos formatos

Los CMS permiten manejar de manera independiente el contenido del diseño, además de gestionar, bajo un formato estandarizado, la información del servidor, reduciendo el tamaño de las páginas para descarga y reduciendo el coste de gestión del portal con respecto a una página estática.

Brindan la posibilidad de escoger diferentes niveles de acceso para los usuarios, ya sea permiso de administración, edición, creador de contenido entre otros.

1.7.1 Drupal

Drupal es un CMS de contenido modular y muy configurable. Este CMS posee licencia GNU/GPL (en inglés GNU General Public License), se destaca por la calidad probada de su código y de las páginas que se generan con el mismo. Respeta de sobremanera los estándares web haciendo énfasis en la usabilidad y consistencia del sistema en general.

Es altamente flexible y adaptable a cualquier entorno de desarrollo, por lo que es multiplataforma. (2011).

A pesar de existir una amplia gama de CMS, se seleccionó Drupal como CMS para el Sistema para la gestión de los procesos de la DCV, debido a que este tiene una activa comunidad en

internet y también dentro de la Universidad, la cual es muy dinámica y altamente efectiva en el mantenimiento y desarrollo de módulos que pueden ser utilizados efectivamente.

Es además un CMS compuesto por módulos, los cuales son altamente reutilizables, lo que reduce el tiempo de desarrollo. Es de fácil entendimiento por los desarrolladores, por lo que el tiempo de capacitación es relativamente corto.

Posee un módulo dedicado estrictamente a la seguridad, esto le brinda al sistema los niveles de seguridad requeridos por el cliente, aumentando así las expectativas del mismo.

Se estará haciendo uso de la última versión estable publicada el 25 de mayo de 2011, versión 7.2.

1.7.2 WordPress

WordPress es una poderosa plataforma de gestión de contenido diseñada para un fácil manejo, incluso para aquellos que no son profesionales en el ramo de la informática. Este CMS es de distribución libre conforme a la licencia estándar GNU/GPL.

Con esta plataforma se logran excelentes resultados en la web respetando todos sus estándares, tiene el don de ser ligero, rápido y gratis. Posee una configuración intuitiva y características bien pensadas, además de un núcleo extremadamente personalizable.

Para lograr que funcione este CMS de forma óptima se deberá poseer un servidor que soporte PHP (versión 4.2 o superior), y como gestor de base de datos MySQL (versión 3.23.x o superior). Puede ser instalado en un servidor web dedicado o compartido, permitiendo que su bitácora sea completamente accesible a quien posea los privilegios de administración.

Permite ahorrar ancho de banda activando la opción de compresión gzip en las opciones de WordPress. Esto facilita el tráfico de información por la red, sobre todo si la conexión con que se cuenta no es de alta velocidad y no posee un ancho de banda bondadoso.

Para la seguridad, el sistema implementa acceso por niveles de usuarios, lo cual protege las secciones más vulnerables contra intrusos. Cuenta con una amplia comunidad de usuarios en internet que actualizan y mejoran constantemente las extensiones o plugins, lo cual extiende las funcionalidades del mismo.

1.8 Ingeniería de Requisitos

La actividad de análisis es una de las principales en el proceso de desarrollo de software y consiste en investigar el problema que da origen, interiorizarlo y describirlo, para luego obtener las

necesidades del cliente o requisitos. (Larman, 1999) En el momento de realizar el análisis es necesario una adecuada comunicación entre los clientes y los analistas, debido que partiendo del intercambio que se realiza se obtienen los requisitos a implementar en el sistema.

Para lograr comprender qué es la Ingeniera de Requisitos, es necesario definir qué es un requisito y cuál es su función. A continuación se mencionan algunos de los conceptos existentes actualmente:

- Es una condición o una capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal. (IEEE, 1990)
- Es una descripción de los servicios que brindará el sistema y las restricciones que este incluye. Representa la necesidad que tienen los clientes de un sistema que les resuelve determinado problema. (Sommerville, 2005)

Los requisitos de software se clasifican en dos grupos: los funcionales y los no funcionales. Los requisitos funcionales determinan una capacidad o condición que debe tener el sistema y los no funcionales son una propiedad o cualidad que el producto debe tener, o sea, características que hacen el producto atractivo, rápido, confiable etc.

La Ingeniería de Requisitos es la aplicación de los procedimientos, técnicas, lenguajes y herramientas para obtener como resultado el análisis, la documentación correspondiente y el seguimiento de las necesidades del usuario. Es un proceso que incluye el descubrimiento de lo que el usuario quiere, el refinamiento de estos, su modelado y su especificación. También se realiza un análisis detallado de las soluciones que se proponen. (Pressman, 2001)

La Ingeniería de Requerimientos tiene un papel muy importante en la construcción de un software, pero a su vez es uno de los procesos más complejos, puesto que según Frederick P. Brooks "*La parte más difícil de construir un sistema es precisamente saber qué construir. Ninguna otra parte del trabajo conceptual es tan difícil como establecer los requerimientos técnicos detallados… Ninguna otra parte del trabajo afecta tanto el sistema si es hecha mal. Ninguna es tan difícil de corregir más adelante… Entonces, la tarea más importante que el ingeniero de software hace para el cliente es la extracción iterativa y el refinamiento de los requerimientos del producto*"

Su objetivo principal es lograr que se generen las especificaciones de las necesidades de los usuarios o clientes de forma correcta, con claridad, sin ambigüedades, en forma consistente y compacta.

1.9 Modelo de Sistema

El modelo de sistema, describe cada una de las funcionalidades que se proponen para el sistema. Incluye el diagrama de casos de uso con sus descripciones.

Un caso de uso representa la interacción entre un usuario, que puede ser una persona o una máquina, y el sistema que se desea obtener. Cada uno de estos casos de uso que van incluido en el modelo de casos de uso, tiene una descripción asociada, que describe las funcionalidades que engloba.

Un caso de uso puede incluir o extender otros, con el objetivo de complementar funcionalidades. Por lo general, los casos de uso se relacionan con actores, que son los que inicializan las acciones que describen.

La descripción de un caso de uso incluye el nombre del caso de uso, los requisitos que engloba, las restricciones que definen lo que se puede o no hacer, los actores que interactúan con el sistema y la descripción de los casos de uso con los diferentes escenarios que facilitan el entendimiento de la descripción. (Una introducción al UML. El Modelo de Casos de Uso.)

El actor es considerado como una entidad fuera del sistema que interactúa con el caso de uso y participa en la historia que describe este. Por lo general, realiza peticiones que funcionan como eventos de entrada o recibe algo del sistema.

1.9.1 Patrones de Casos de usos

Un patrón es un problema/solución que estandariza principios y sugerencias relacionadas frecuentemente con la asignación de responsabilidades. Es la representación de reiterados problemas en un tema determinado. (Use Cases: Patterns and Blueprints, 2004)

Para la realización del diagrama de Casos de Uso (CU) se utilizan un conjunto de patrones que facilitan la representación de estos. Dentro de los principales patrones de CU están:

Regla del negocio: Se fundamenta en la extracción de información relacionada con el cliente (políticas, reglas y regulaciones del negocio) que luego son tratadas como reglas del negocio.

Concordancia:

Reusabilidad: Es separar en un caso de uso la sub-secuencia de acciones que son obligatorias para más de un caso de uso.

..

Adición: Es separar en un caso de uso la sub-secuencia de acciones que son alternativas a las funcionalidades de otros casos de uso.

Especialización: Es la representación de una herencia pero enmarcada en casos de uso. Varios casos de uso hijos son del mismo tipo de un padre con funcionalidades genéricas.

CRUD:

Completo: Representa a un conjunto de funcionalidades que engloban relación. Maneja operaciones como Creación, Lectura, Actualización y Eliminación (por sus siglas en inglés Create, Read, Update, Delete CRUD). Por lo general son llamados CRUD o Gestionar.

Parcial: Son similares a los CRUD Completos, pero separan una funcionalidad por determinadas características que la hacen compleja o tediosas.

Múltiples actores:

Roles diferentes: Consiste en un caso de uso y por lo menos dos actores. Es utilizado cuando dos actores juegan diferentes roles en un caso de uso, o sea, interactúan de forma diferente con el caso de uso.

Roles común: Se aplica cuando dos actores juegan un mismo rol sobre un caso de uso. Representa la herencia entre actores.

1.10 Diseño de Sistemas

El diseño de sistemas de software se realiza para transformar los requisitos identificados en un diseño detallado del sistema en cuestión. Tiene como finalidad evolucionar hacia una arquitectura sólida para el sistema y generar un diseño que sea entendible por los desarrolladores y que ayude y facilite la implementación. Cuando se realiza el diseño de un sistema, se tienen en cuenta un grupo de elementos que amplían y describen el comportamiento del sistema, facilitando el entendimiento de cómo se deben ejecutar las funcionalidades. A continuación se describen algunos de ellos, como los patrones de diseño, diagramas de secuencia y diagramas de clases del diseño.

1.10.1 Patrones de Diseño

Los patrones de diseño son propuestas de soluciones a problemas frecuentes surgidos en el proceso de desarrollo de software, estos brindan una solución a problemas similares que está probada y bien descrita, o sea, que están sujetos a situaciones muy parecidas. Un patrón de

diseño es un ente descriptible de acuerdo a su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) así como sus consecuencias (costos y beneficios).

De forma general, un patrón de diseño se puede definir como:

- Una forma más práctica de describir determinados elementos de la organización de un programa.
- Una solución estándar para un problema común de programación.
- Una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios.
- Un proyecto o estructura de implementación que logra una finalidad determinada.

Actualmente existen muchos patrones de diseño de software, estos a su vez se agrupan en grandes grupos, entre los cuales tenemos:

Patrones GoF (Gant of Four):

Patrones Creacionales: Inicialización y configuración de objetos.

- Abstract Factory: Proporciona una interfaz para crear familias de objetos relacionados o dependientes sin especificar su clase concreta.
- Builder: Permite a un objeto construir un objeto complejo especificando sólo su tipo y contenido.
- Factory Method: Define una interfaz para crear un objeto dejando a las subclases decidir el tipo específico al que pertenecen.
- Prototype: Permite a un objeto crear objetos personalizados sin conocer su clase exacta a los detalles de cómo crearlos.
- Singleton: Garantiza que solamente se cree una instancia de la clase y provee un punto de acceso global a la misma.

Patrones Estructurales: Separan la interfaz de la implementación. Se ocupan de cómo las clases y objetos se agrupan, para formar estructuras más grandes.

- Adapter: Convierte la interfaz que ofrece una clase en otra esperada por los clientes.
- Bridge: Desacopla una abstracción de su implementación y les permite variar independientemente.

- Composite: Permite gestionar objetos complejos e individuales de forma uniforme.
- Decorator: Extiende la funcionalidad de un objeto dinámicamente de tal modo que es transparente a sus clientes.

Patrones de Comportamiento: Más que describir objetos o clases, describen la comunicación entre ellos.

- Chain of Responsability: Evita el acoplamiento entre quien envía una petición y el receptor de la misma.
- Command: Encapsula una petición de un comando como un objeto.
- Interpreter: Dado un lenguaje define una representación para su gramática y permite interpretar sus sentencias.
- Iterator: Acceso secuencial a los elementos de una colección.
- Mediator: Define una comunicación simplificada entre clases.
- Memento: Captura y restaura un estado interno de un objeto.
- Observer: Una forma de notificar cambios a diferentes clases dependientes.

Se recomienda de forma general usar patrones de diseño solo en aquellos casos en los que se tiene un total dominio del problema, utilizando además un algoritmo eficiente desde el comienzo y a lo largo de todo el ciclo de desarrollo. Los patrones de diseño pueden aumentar o disminuir la capacidad de entendimiento de un diseño o de una implementación, pueden hacer lo mismo con la cantidad de código, todo depende del uso adecuado que se les dé. Una vez dominados los patrones de diseño, éstos serán de gran ayuda a la hora de incorporar calidad al producto que se desarrolla, así como valor agregado al mismo.

Los patrones de diseño deben estar al alcance de todos los desarrolladores de software, difundir la experiencia adquirida por todos los miembros del equipo suele ser una ventaja, pues se generaliza el conocimiento y se agiliza de sobremanera el desarrollo.

1.10.2 Modelo del Diseño

El modelo de diseño es un artefacto que se genera en el flujo de trabajo Análisis y Diseño, específicamente en la etapa de diseño. Este constituye una abstracción de la implementación del sistema y tiene como objetivo documentar el diseño sobre el que se apoya el flujo de trabajo de

implementación. Está compuesto por clases de diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos.

Los paquetes de diseño son los encargados de agrupar los elementos de modelo de diseño relacionados con el objetivo de darle organización. No ofrecen una interfaz formal, aunque puede darse el caso de que expongan algunos de sus contenidos que ofrecen comportamiento. Deben utilizarse fundamentalmente como herramienta organizativa del modelo, para agrupar elementos relacionados.

Los subsistemas, a diferencia de los paquetes de diseño, encapsulan comportamiento, proporcionando interfaces formales y no incluyen el contenido interno.

Las clases del diseño son una representación de un grupo de objetos que tienen la misma responsabilidades, relaciones, operaciones y atributos. (Microsoft, 2007)

1.11 Métricas para el diseño

Las métricas en cualquier ciencia tienen una gran importancia, pues permiten medir la calidad con que se obtienen los resultados esperados. La ingeniería informática no se queda atrás en este sentido, se deben aplicar métricas que permitan medir con objetividad no con subjetividad la calidad del producto, las líneas de código producidas, velocidad de ejecución, tamaño de memoria, y los defectos informados durante un periodo de tiempo establecido. Se deben recopilar las métricas acorde a lo que se desea medir para que los indicadores del proceso y del producto puedan ser ciertos.

1.11.1 Métrica de acoplamiento

Dhama define que el acoplamiento demuestra el nivel de conectividad de un módulo con otro módulo, datos globales y el entorno exterior.

Para el acoplamiento de flujo de datos y de control:

d(i) = número de parámetros de datos de entrada.

c(i) = número de parámetros de control de entrada.

d(o) = número de parámetros de datos de salida.

c(o) = número de parámetros de control de salida.

Para el acoplamiento global:

g(d) = número de variables globales usadas como datos

g(c) = número de variables globales usadas como control.

Para el acoplamiento de entorno:

w = número de módulos llamados (expansión).

r = número de módulos que llaman al módulo en cuestión (concentración).

Usando estas medidas, se define un indicador de acoplamiento de módulo, (*m(*c)), de la siguiente manera: $m(c) = k/M$; donde k es una constante de proporcionalidad con valor igual 1.

$$M = d(i) + a\,x\,c(i) + d(o) + b\,x\,c(o) + g(d) + c\,x\,g(c) + w + r$$

Donde a = b = c = 2.

Cuanto mayor es el valor de *m(c)*, menor es el acoplamiento de módulo. (Pressman, 2001)

1.11.2 Métrica de cohesión

Bieman y Ott definen que las métricas de cohesión proporcionan una indicación de cohesión de un módulo y tienen un valor que varía de 0 a 1. Asumen un valor de 0 si el procedimiento tiene más de una salida y no muestra ningún atributo de cohesión que ha sido indicado por alguna métrica. Un procedimiento sin muestras de súper-unión, sin muestras comunes a todas las porciones de datos, no tiene una cohesión funcional fuerte (no hay señales de datos que contribuyan a todas las salidas). A continuación de describe el carácter de esta métrica para la cohesión funcional fuerte.

$$CFF\ (i) = SU\ (SA\ (i))/\ muestra\ (i)$$

Donde SU (SA (i)) denota muestra de súper-unión (el conjunto de señales de datos que se encuentran en todas las porciones de datos de un módulo i). Como la relación de muestras de súper-unión con respecto al número total de muestras en un módulo i aumenta hasta un valor máximo de 1, la cohesión funcional del módulo también aumenta. (Pressman, 2001)

1.11.3 Métricas orientadas al tamaño de la clase

El tamaño general de una clase se puede calcular mediante los siguientes planteamientos:

- El número total de operaciones (tanto operaciones heredadas como operaciones privadas de la instancia) que se encuentran encapsuladas dentro de la clase.
- El número de atributos (tanto atributos heredados como atributos privados de la instancia)

que se encuentran encapsulados en la clase.

Si los valores obtenidos son demasiado grandes entonces se asume que la clase puede tener demasiada responsabilidad, lo cual provoca la reducción de la reutilización de la clase y se tornará complicada la implementación y la prueba. Contrario a esto ocurre si los valores TC son de menor valor.

También es necesaria una evaluación concreta de las métricas mediante los umbrales. Algunos especialistas plantean la siguiente clasificación:

Clasificación	**Valores de los umbrales**
Pequeño	<= 20
Medio	>20 <= 30
Grande	>30

Tabla 1 Umbrales para la clasificación del tamaño de clase

Finalmente se calcula los promedios correspondientes a los diferentes valores para tener una estimación general del sistema.

1.12 Métodos para medir la satisfacción del cliente

La adecuada captura de requerimientos es esencial para que un proyecto de software sea exitoso, sobre todo si se tiene en cuenta que esta es una difícil tarea, la cual requiere la puesta en práctica de una amplia variedad de estrategias con el fin de extraer del cliente todo lo que este necesita y contribuir así a su satisfacción.

Existen varios métodos para medir la satisfacción del cliente, a continuación se abordan 2 de ellos, haciendo énfasis en el método de Kano.

1.12.1 Modelo ACSI de Satisfacción del cliente

The American Customer Satisfaction Index (ACSI) es un indicador que establece el nivel de satisfacción de los ciudadanos de los EEUU con los productos y servicios recibidos desde 1994.

La representación gráfica de este modelo se presenta a continuación:

Figura 2 Diagrama del modelo ACSI

El valor del indicador se obtiene del tratamiento de las respuestas de los estadounidenses a un cuestionario telefónico, y se presentan los resultados en cuatro niveles (American Customer Satisfaction Index Homepage):

- Valor del indicador a nivel nacional.
- Valor del indicador en diez sectores económicos.
- Valor del indicador en cuarenta y tres industrias diferentes.
- Valor del indicador en más de doscientas empresas y agencias del gobierno.

Expectativas del cliente: las expectativas del cliente son una medida anticipada de la calidad que el cliente espera recibir por los productos y servicios que la organización ofrece.

Calidad percibida: tomando como entrada las expectativas del cliente, la calidad percibida se considera asociada principalmente a dos factores: la personalización y la fiabilidad.

Valor percibido: este parámetro expresa la relación entre la calidad obtenida y el precio pagado.

Quejas del cliente: las quejas son la expresión más palpable de la insatisfacción. Cuanto más satisfecho está un cliente, menos ganas tiene de expresar una queja.

Fidelidad del cliente: la fidelidad del cliente es el componente crítico del modelo. Se observa que, si bien la satisfacción del cliente ocupa un lugar central en el diagrama, las flechas relacionales desembocan en este parámetro.

1.12.2 Método de Kano

Este método consiste en un modelo de evaluación de la calidad de los productos, el cual permite medir la relación entre la funcionalidad de un producto x y la satisfacción que esta funcionalidad le brinda a los clientes. Este método fue definido en la Universidad de Tokio por un académico de la misma nombrado Noriaki Kano. El método permite además discriminar y clasificar los requisitos en tres tipos ideales de atributos en función de la relación entre funcionalidad y satisfacción como se muestra a continuación. (León Duarte, 2005)

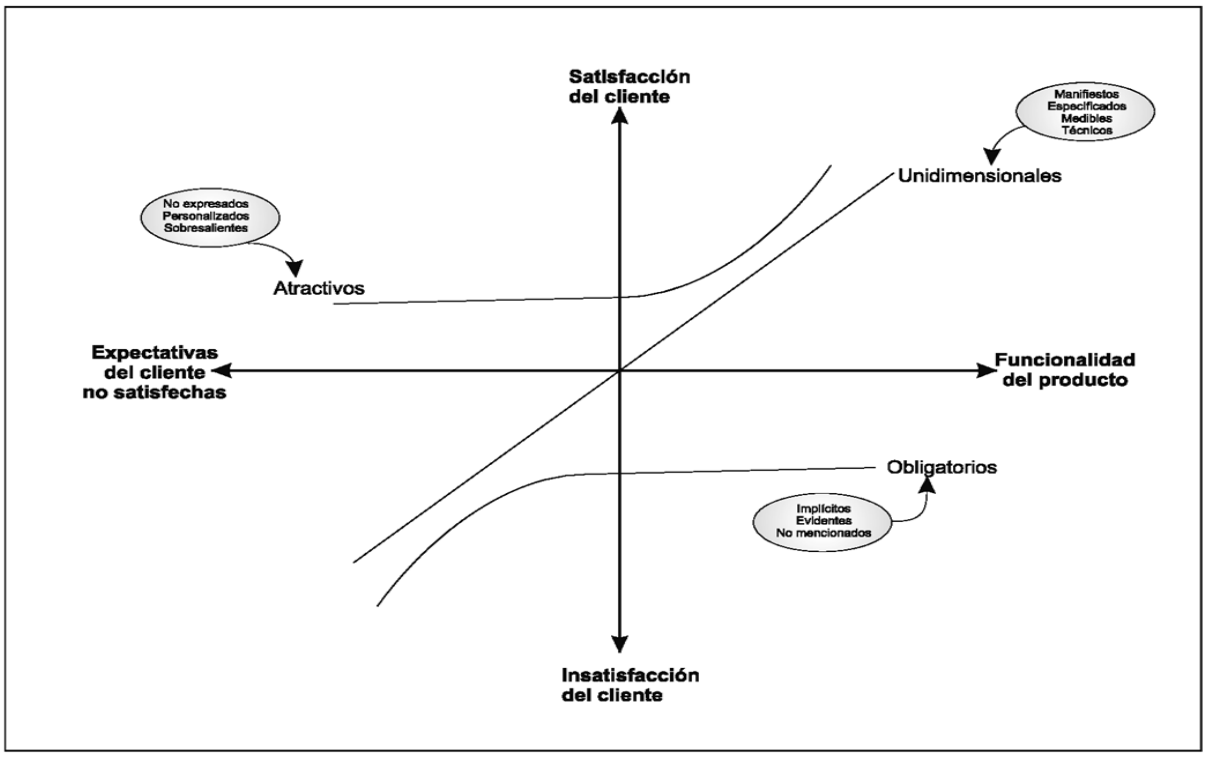


Figura 3 Clasificación de los requisitos con el método de Kano

Atractivos: son aquéllos que, por debajo de cierto umbral de funcionalidad, mantienen un nivel de satisfacción relativamente bajo y constante, pero que, una vez superado ese umbral, producen un aumento significativo de la satisfacción. Los requisitos atractivos suelen denominarse deleitosos (del inglés delighter).

Unidimensionales: se caracterizan porque la satisfacción que producen aumenta de modo aproximadamente proporcional al nivel de funcionalidad. Responden a la percepción tradicional de la relación entre funcionalidad y satisfacción: a mayor funcionalidad, se observa una mayor satisfacción.

Obligatorios: son aquéllos que, hacia las gamas bajas de funcionalidad, aumentan la satisfacción en relación directa con la funcionalidad pero que, superado cierto umbral, dejan de producir un incremento importante en la satisfacción.

Para obtener la clasificación antes mencionada Kano (Kano, 1984) propone un cuestionario, en el que se formulan dos preguntas por cada requerimiento analizado:

- ¿Cómo se siente si la característica X está presente en el producto? (Requisitos funcionales)
- ¿Cómo se siente si la característica X NO está presente en el producto? (Requisitos disfuncionales)

Para cada pregunta el cliente responde entre cinco posibles opciones:

1. Me agrada
2. Es de esperarse
3. Neutral
4. Lo acepto
5. Me desagrada

Una vez respondidas las dos preguntas se busca su combinación en la Tabla de Clasificación de los Requisitos, y de esta forma los requisitos de un producto se pueden clasificar en una de las seis categorías que se muestran a continuación:

A: Atractivo

B: Obligatorio

U: Unidimensional

I: Indiferencia

Inv: Respuesta inversa

D: Respuesta dudosa

Cuando un cliente se comporta Indiferente indica que una mayor o menor funcionalidad respecto a este requerimiento no se refleja en un aumento o disminución de la satisfacción del cliente y se representaría como una recta paralela al eje horizontal de la Figura 2. Una respuesta Inversa indica que la interpretación de criterios funcionales y disfuncionales del diseñador es la inversa a la percepción del cliente, es decir, lo que la pregunta supone como funcional es percibido como no funcional por quien responde. Por último, cuando existe una contradicción en las repuestas a las preguntas, se clasifica como una respuesta Dudosa. Esto quiere decir que no es razonable contestar lo mismo ante un par de preguntas complementarias.

		Requisitos disfuncionales				
Requisitos		**1**	**2**	**3**	**4**	**5**
funcionales	**1**	D	A	A	A	U
	2	Inv	I	I	I	O
	3	Inv	I	I	I	O
	4	Inv	I	I	I	O
	5	Inv	Inv	Inv	Inv	D

Figura 4 Clasificación de los Requisitos

Una vez analizados los métodos para medir la satisfacción del cliente antes expuesto, se decidió utilizar el método de Kano, por ser el más preciso en la relación satisfacción - funcionalidad que es precisamente lo que se desea. Es además un método de fácil aplicación y poco costoso, por lo que aumenta su idoneidad en la investigación.

1.13 Conclusiones

En este capítulo se hizo un estudio detallado de los principales aspectos relacionados con el Análisis y Diseño de un sistema. Se hizo una comparación de las principales herramientas de gestión de proyecto usadas en el mundo, Cuba y específicamente en la UCI. También se realizó un estudio de las tendencias mundiales sobre metodologías de desarrollo, lenguajes de modelado y las herramientas CASE que dan soporte a estos. Como resultado del estudio y comprensión de los elementos anteriormente mencionados se arriba a las siguientes conclusiones:

- Las herramientas Project y Redmine no satisfacen las necesidades de los trabajadores de la DCV, por tener los clientes de la presente investigación un flujo de procesos y actividades que no se ajustan a las que soportan las herramientas de gestión de proyecto mencionadas anteriormente.

- Se estarán utilizando algunas características que poseen las herramientas de gestión de proyecto, debido a que son universales en la gestión de proyecto, y se ajustan perfectamente a lo que se desea lograr.
- Se ha seleccionado la metodología RUP pues es flexible y altamente adaptable a las necesidades de cada proyecto, sin importar cuán grande o pequeño este sea. Brinda la documentación necesaria para que la investigación continúe y perdure en el tiempo, sin perder de vista los objetivos principales.
- Se utilizará el lenguaje de modelado de negocio BPMN por ser un lenguaje muy descriptivo y fácil de modelar, que permite la clara comprensión de la estructura de una empresa.
- Se hará uso del lenguaje de modelado de sistema UML por ser el lenguaje que propone la metodología RUP. Tiene probadas ventajas que se aprovecharan en este trabajo.
- Se usará como lenguaje de programación del lado del servidor PHP, por sus grandes prestaciones para la generación de web dinámicas. Es además un lenguaje multiplataforma, no privativo y de fácil asimilación.
- Se empleará como herramienta CASE Visual Paradigm, por ser una herramienta muy factible para el modelado, da soporte a varios lenguajes de modelado como BPMN y UML.
- El análisis es una importante etapa dentro del proceso de desarrollo de software, este permite entender y describir cómo funciona la empresa cliente y determinar las características del sistema que se desea realizar.
- El diseño también es muy importante porque es la base de la implementación, que permite modelar a un alto nivel de detalles cómo será la interacción entre clases, componentes y objetos del sistema.
- Se hará uso de método de Kano para la medición de satisfacción del cliente, por la exactitud que este brinda en este sentido.

Capítulo 2: Modelado del negocio. Análisis y diseño del sistema

2.1 Introducción

En este capítulo se describe cómo fue concebido el sistema para la gestión de los procesos de la DCV en términos del análisis y el diseño. Se explican los procesos por los que está compuesta la DCV, la interacción entre ellos y su representación. Se detalla todo el flujo de trabajo de levantamiento de requisitos, modelo de Casos de uso y la descripción que engloban. Por último, se especifica y se muestra el diseño realizado como etapa anterior a la implementación del sistema.

2.2 Modelado de procesos

El modelado de procesos es muy importante en el proceso de desarrollo de software ya que les permite a los miembros del proyecto entender claramente el funcionamiento de la empresa. Su principal objetivo es detectar los problemas actuales que presenta esta y brindar posibles mejoras. El modelado de procesos en general, permite obtener una visión estática de la estructura de la organización.

2.2.1 Procesos de la Dirección de Comunicación Visual

La Dirección de Comunicación Visual es la entidad en la Universidad de las Ciencias Informáticas que se encarga de realizar trabajos visuales en las modalidades de Diseño, Realización y Producción. Todos estos servicios se combinan mediante una secuencia de procesos y actividades que están estrechamente relacionadas y que se ejecutan para obtener la propuesta final que es entregada al cliente.

El proceso general que engloba los sub-procesos de la DVC se describe desde el momento en que el cliente realiza una solicitud a la DCV, que luego se le da solución en el sub-proceso encargado de confeccionar el entregable, para posteriormente pasarla al grupo de calidad y decidir si está listo o no para entregárselo al cliente. El diagrama de procesos se muestra a continuación:

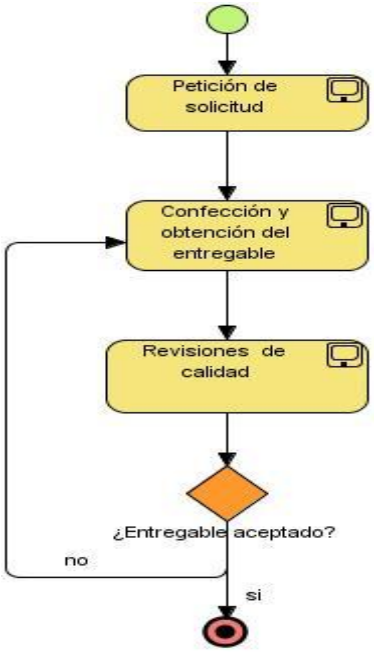


Figura 5 Diagrama de procesos de la DCV

2.2.1.1 Sub-proceso "Petición de solicitud":

El sub-proceso "Petición de solicitud" está compuesto por las siguientes actividades:

Actividad "Petición de solicitud":

El cliente de la DCV es el encargado de hacer la solicitud a la entidad. El cliente puede ser un proyecto de producción de software, otro sector específico de la universidad como la rectoría o una entidad externa a la universidad. En caso que la solicitud la realice un proyecto productivo entonces existen dos clientes involucrados en una misma solicitud, pues la DCV da respuesta a las solicitudes hechas por el proyecto que a su vez responde a las necesidades del cliente del proyecto (cubano o extranjero). En este caso los dos (Cliente del proyecto y el Proyecto) son clientes de la DCV. La solicitud al realizarse incluye un grupo de datos que sirven de previo conocimiento al director de la DCV. Esta actividad tiene como salida la solicitud de diseño.

Actividad "Revisión de la solicitud":

Una vez que se realiza una solicitud, pasa directamente al director de la DCV que es el encargado de revisarla. Esta tiene como objetivo hacer un análisis minucioso de las características del proyecto al que se le quiere brindar el servicio para posteriormente decidir el momento y el lugar donde se debe realizar la reunión de inicio. Tiene como entrada la solicitud de diseño.

Actividad "Reunión de inicio":

La reunión de inicio se hace en la fecha y el lugar establecido por el director de la DCV y se realiza con el objetivo de conocer un poco más sobre las características del proyecto. Este proceso tiene como entrada la solicitud de diseño realizada en el proceso de Petición de solicitud. Una vez concluida la reunión se le da una respuesta al cliente donde puede aceptarse o denegarse la solicitud. Si la solicitud es denegada, se describe claramente el motivo para luego pasar a ser archivada. En caso que sea aceptada puede que no se le pueda dar solución en un tiempo cercano por lo que la solicitud pasa a un estado de espera, en otro caso (aceptada sin estado de espera) se pasa a la actividad de asignación de proyecto.

Actividad "Asignación de proyecto":

Un proyecto puede ser asignado por el director de la DCV, jefes de departamento y jefes de grupo a sus subordinados o miembros del equipo de trabajo. Tiene como entrada la solicitud de diseño y como salida el proyecto asignado.

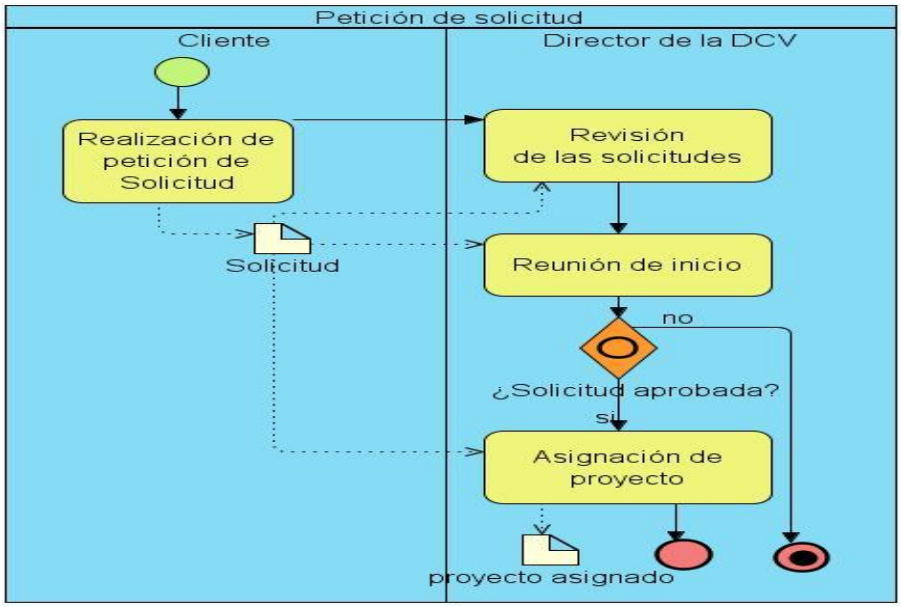


Figura 6 Sub-proceso "Petición de solicitud"

2.2.1.2 Sub-proceso "Confección y obtención del entregable":

El sub-proceso "Confección y obtención de entregable" está compuesto por las siguientes actividades:

Actividad "Creación del cronograma":

El trabajador al que se le asigne el proyecto es el responsable de crear el cronograma correspondiente. Este cronograma se realiza con el objetivo de llevar un control estricto de las actividades que se realizan para dar respuesta a una solicitud. Este cronograma puede ser afectado en más de una ocasión y tiene como entrada el proyecto que le fue asignado al trabajador de la DCV y como salida el cronograma del proyecto.

Actividad "Aprobación del cronograma de trabajo":

Una vez hecho el cronograma del proyecto, el jefe inmediato superior lo revisa y decide si está acorde con el tiempo que ha sido estimado para la realización de la propuesta de trabajo. En caso de no estar de acuerdo, el diseñador debe pasar a estructurar nuevamente el cronograma. Si está de acuerdo con la propuesta de tiempo, entonces se pasa a mostrárselo al cliente. Esta actividad tiene como entrada el cronograma realizado en la actividad anterior.

Actividad "Visualización del cronograma":

Una vez aprobado el cronograma, se le muestra al cliente a ver si está de acuerdo con el tiempo establecido para la realización de las actividades que involucra. En caso de no estar de acuerdo el trabajador debe reestructurarlo nuevamente. En caso de estar de acuerdo se prosigue al desarrollo del trabajo. Tiene como entrada el cronograma.

Actividad "Desarrollo del trabajo":

En esta actividad el trabajador de la DCV se encarga de realizar el trabajo que se le asignó, desarrollando cada una de las actividades que involucra. Debe desempeñarla acorde al tiempo establecido en el cronograma de trabajo. Tiene como salida un entregable parcial o total del trabajo.

Actividad "Realizar solicitud a calidad":

Una vez terminado parcial o totalmente el producto, se debe hacer una solicitud a calidad interna de la DCV para su posterior revisión. Tiene como salida la solicitud que se le hace al equipo de calidad.

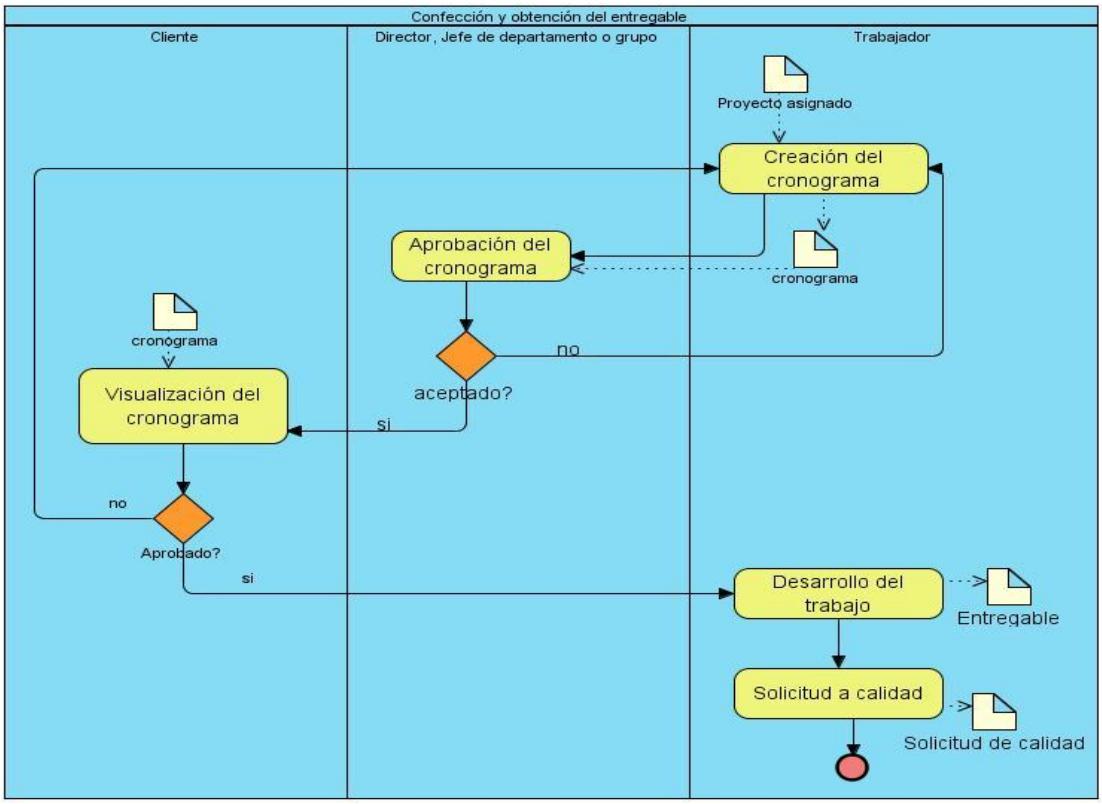


Figura 7 Sub-proceso " Confección y obtención del entregable"

2.2.1.3 Sub-proceso "Revisión de calidad":

El sub-proceso "Revisión de calidad" está compuesto por las siguientes actividades:

Actividad "Elaboración y asignación de actividades":

Consiste en confeccionar una actividad de revisión y asignársela a un miembro del equipo de calidad interna de la DCV luego de que se haga la solicitud a dicho equipo. Este proceso tiene como entrada la solicitud hecha en el proceso de Realizar solicitud de calidad.

Actividad "Revisar producto entregable":

Consiste en hacer una revisión total o parcial del producto. Se realiza parcialmente cuando se concluye alguna sección del trabajo y se desea verificar la calidad. La revisión total se realiza cuando se quiere verificar la integración de las partes que han sido revisadas anteriormente. En caso de ser aceptado el producto en la revisión parcial se prosigue a continuar con la realización de este. En caso de ser aceptada la revisión del producto final pasa a ser mostrado al cliente para su aceptación y en caso de ser rechazado tanto la revisión total como la parcial se pasa a una

restructuración del cronograma para poder ejecutarla nuevamente. Tiene como entrada el entregable que se obtiene luego de ejecutar el proceso ¨Desarrollo del trabajo¨.

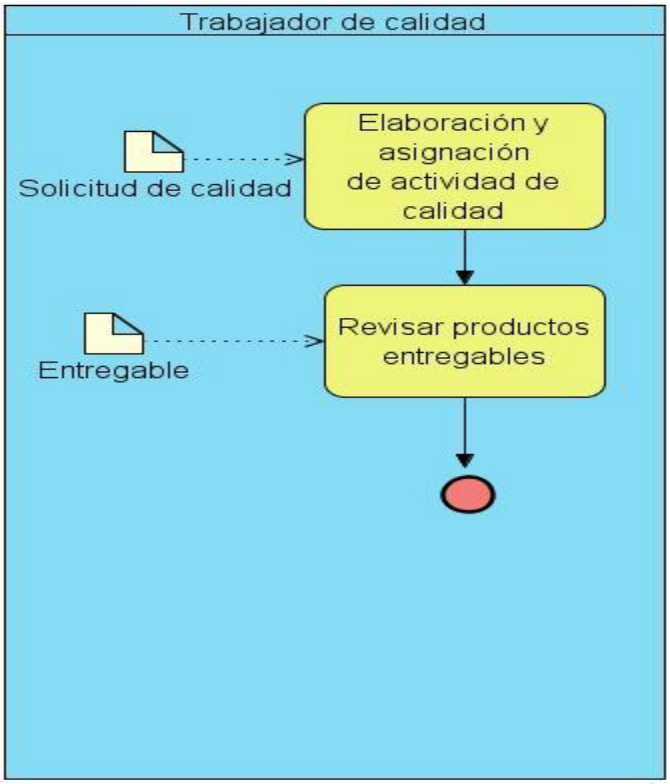


Figura 8 Revisión de calidad

2.3 Requisitos del sistema

Los requisitos del sistema son el resultado del análisis detallado de las actividades automatizables del negocio. Estos representan las funcionalidades que componen el sistema y sus características. A continuación se muestra una tabla resumiendo la cantidad de requisitos elicitados con sus respectivos niveles de prioridad y complejidad. Para más información dirigirse al documento Especificación de Requisitos.

2.3.1 Requisitos funcionales

Requisitos funcionales	Complejidad			Importancia		
105	Alta	Media	Baja	Alta	Media	Baja

	20	37	48	36	55	14

Tabla 2 Requisitos funcionales: Complejidad e Importancia

La **complejidad** de los requisitos funcionales está dada por la cantidad de requisitos que dependen de ese que se clasifica como complejo, es decir, que el nivel de dependencia de ese requisito con otros especificados por el cliente sea relativamente alto. Esta situación hace engorroso dar cumplimiento a esta funcionalidad, pues se le debe dedicar más tiempo en implementarla y entenderla del todo.

Otro aspecto importante para considerar más o menos complejo a un requisito está en la cantidad de datos que este maneje, las relaciones entre clases que se crean para dar cumplimiento del mismo, así como la cantidad de llamadas a uno o varios métodos que se necesiten para realizar su cometido.

El nivel de **importancia** de un requisito está precisamente en cuán necesario sea este para el cliente, que le aporte mayor beneficio en cuestiones de costo y que resuelva sus necesidades más urgentes. Además un requisito importante es aquel por sus características específicas incide directamente en la definición de la arquitectura de software a emplear.

2.3.2 Requisitos no funcionales

Los requisitos no funcionales que a continuación se describen son los que caracterizan el sistema para la gestión de los procesos de la DCV. Ellos dan soporte a los requisitos funcionales descritos en el documento especificación de requisitos y mencionados en la sección anterior.

Usabilidad

Permitir el uso del teclado

El sistema debe permitir el uso del teclado, debido a que les facilita la labor a aquellas personas que trabajan constantemente con la aplicación y que tienen habilidades sobre este dispositivo de entrada, además, en caso de no contarse con un mouse, se puede hacer uso de este.

Información de forma lógica

La información que se muestre en la aplicación debe estar ubicada de forma lógica y coherente, para poder facilitarles la comprensión de lo que significan los valores de entrada y los datos que muestra como valores de salida.

Disponibilidad desde cualquier plataforma

El sistema debe estar disponible desde cualquier plataforma, debido a que en la universidad se necesitan aplicaciones que sean multiplataforma, además, los clientes (DCV) en muchas ocasiones usan diferentes plataformas para la realización de sus tareas como es el caso de Macintosh y Windows.

Configuración fácil y rápida

El sistema debe permitir realizar las configuraciones de forma fácil y rápida, para garantizar que no se le tornen tediosas, fundamentalmente en la gestión de los nomencladores del sistema.

Disponibilidad desde toda la UCI

El portal debe estar accesible desde toda la UCI, para facilitarle la interacción de los usuarios desde cualquier área de la universidad.

Fiabilidad

Autenticación segura

El sistema debe permitir la autenticación mediante el usuario y la contraseña UCI. Debe ser una autenticación segura, cumpliendo con las políticas de seguridad de la universidad.

Protección de la información sensible

La información sensible que viaja por la red no debe ir en texto plano.

Acceso a la información según el rol

Para acceder a la parte de gestión, el usuario debe haberse autenticado previamente y así poder acceder a las diferentes funcionalidades que tenga permiso.

Cierre de sesión

Dado un tiempo predeterminado sin hacerse uso de la aplicación, debe cerrar la sesión, garantizando de esta manera la no presencia de intrusos en funcionalidades donde no se les ha dado permisos.

Exactitud

Las salidas que ofrece el sistema como resultado del manejo y procesamiento de la información estarán adecuadamente validadas durante los diferentes hitos de prueba, teniendo gran nivel de exactitud con los valores esperados.

Restricciones de diseño

Uso de estándares de codificación

Debe garantizarse el uso de estándares de codificación, para una mejor comprensión de los códigos ejecutados en caso de una segunda versión del sistema.

Requisitos para la documentación de usuarios en línea y ayuda del sistema

Ayuda del sistema

El sistema contará con una ayuda que constituirá una guía de apoyo en el momento de hacer uso de la aplicación. Esta describe todas las funcionalidades del sistema.

Interfaz

Interfaces de usuario

El sistema tendrá las funcionalidades principales en una primera (principal) interfaz. El objetivo es no tener que acceder a varias secciones para ver los elementos que más interés le causen. Debe ser un sistema con las funcionalidades disponible desde un bandeja de entrada con los elementos fundamentales para desempeñar su tarea.

2.4 Modelo de sistema

El modelo de sistema es un artefacto generado con el objetivo de identificar los actores del sistema, los casos de uso con los que interactúan y la descripción de estos. El modelo de sistema muestra los siguientes diagramas de casos de uso, el primero representa la relación de los casos de uso de gestión de los procesos de la DCV y el segundo los casos de uso de configuración y seguridad.

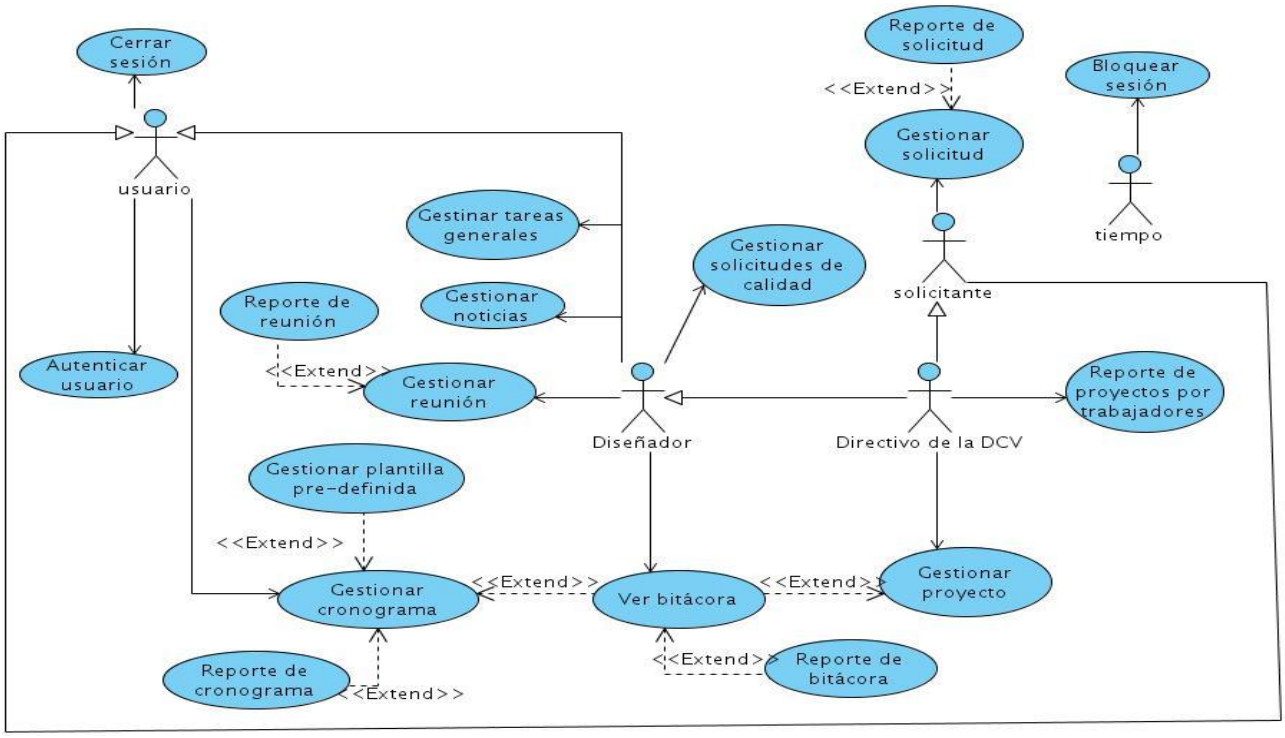


Figura 9 Modelo de CU (Gestión de los procesos de la DCV)

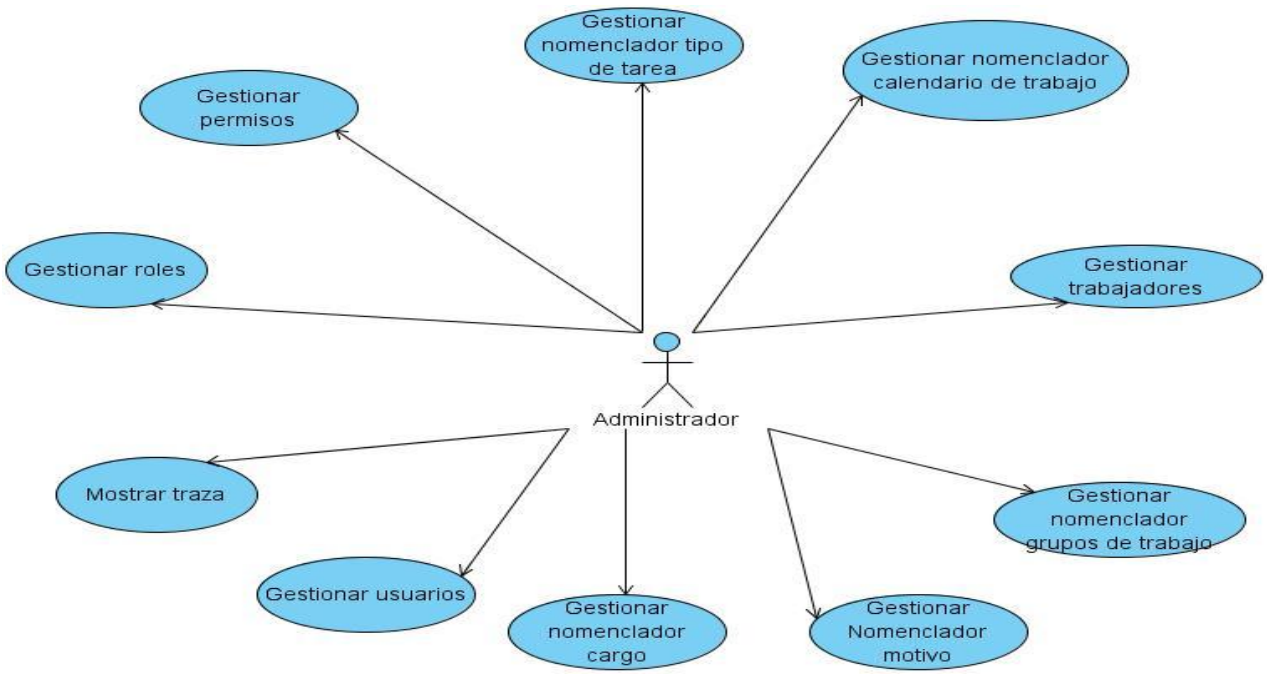


Figura 10 Modelo de CU (Configuración y seguridad)

2.4.1 Actores del sistema

Los actores del sistema son aquellas personas, grupo de personas, entidades o sistemas que van a interactuar con la aplicación. Tienen como objetivo hacer una representación de todo lo que intercambiarán información con el sistema para entender su importancia y delimitar las funcionalidades a las que tendrá acceso. En la siguiente tabla se describen los actores del sistema:

Actor	Descripción
Solicitante	Es la persona que se encarga de realizar una solicitud a la DCV.
Usuario	Son todas las personas que interactúan con el sistema.
Directivo de la DCV	Son los ejecutivos que dirigen y controlan la DCV (Director, Jefe de departamento, Jefe de grupo).
Diseñador	Son los trabajadores que conforman los diferentes grupos de trabajo.
Tiempo	Es un reloj que determina cuando debe cerrarse la aplicación.

Tabla 3 Descripción de los actores del negocio

2.4.2 Patrones de Casos de Uso empleados

Multiples actors: Roles Común

En este caso se representa la especialización de los actores. En su forma más general juegan un mismo papel sobre los casos de uso en que inciden.

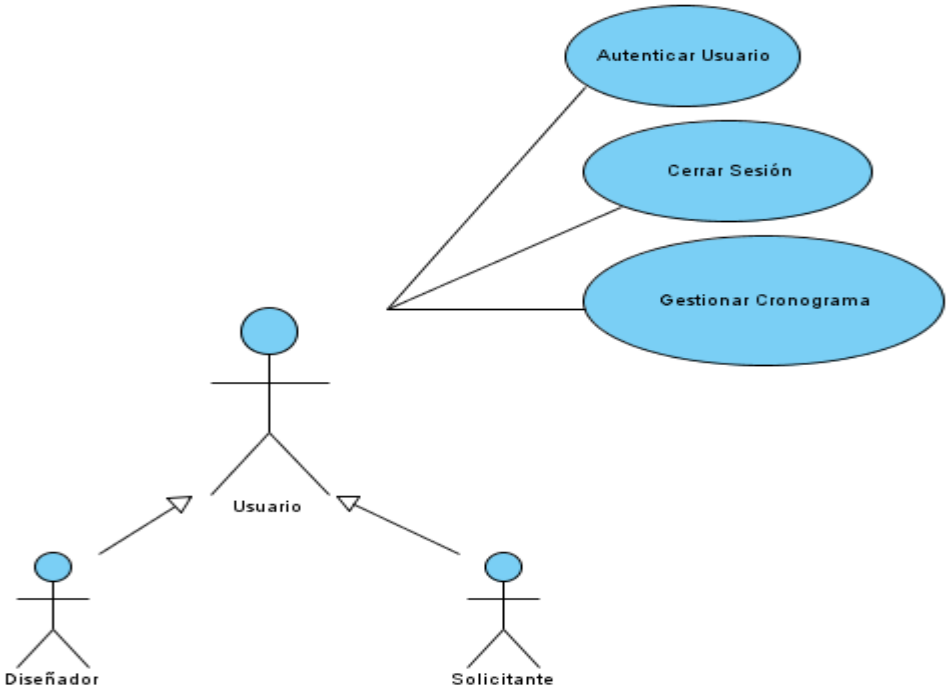


Figura 11 Patrón de CU ¨Múltiples Actores: Roles común¨

CRUD Completo

Representa todo el conjunto de funcionalidades que engloban la relación. Maneja las operaciones de Creación, Lectura, Actualización y Eliminación (por sus siglas en inglés Create, Read, Update, Delete CRUD).

Se muestra un ejemplo de esta situación en 4 casos de uso, sin embargo, está presente en otros casos de uso no mencionados.

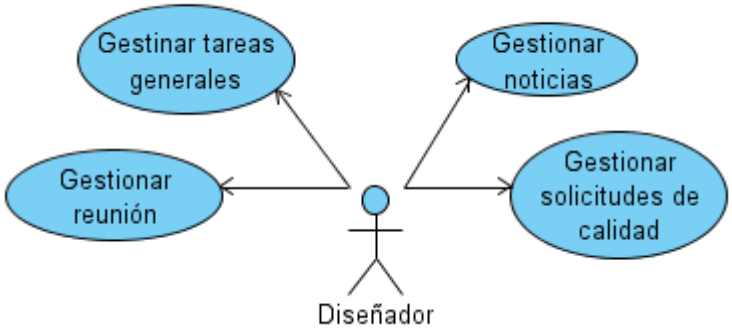


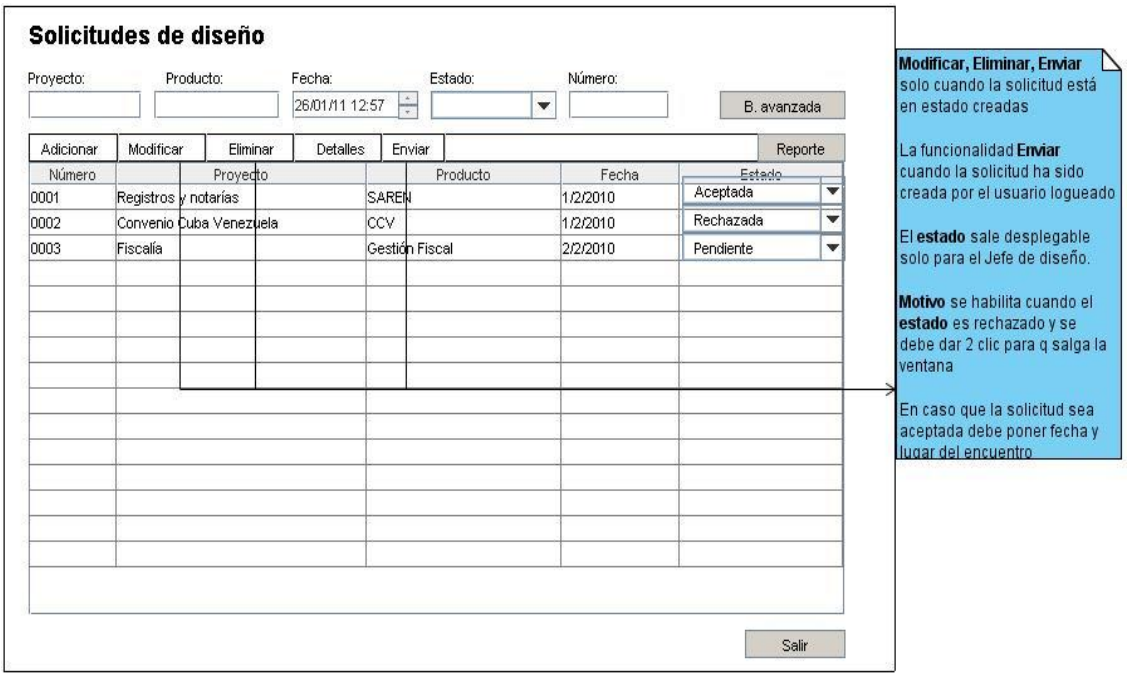
Figura 12 Patrón de CU ¨CRUD Completo¨

2.4.3 Descripción de CU

Los casos de uso críticos son los arquitectónicamente significativos, o sea, que describen las principales funcionalidades de un sistema. A continuación se ejemplifican algunos de los más importantes para el sistema para la gestión de los procesos de la DCV:

Caso de Uso	Gestionar Solicitud
Actores:	Solicitante
Resumen:	Este caso de uso tiene como objetivo que el solicitante pueda realizar una solicitud a la DCV y gestionar la misma, en dependencia del reglamento
Precondiciones	El solicitante debe haberse autenticado
Referencias	RF_001, RF_002, RF_003, RF_004, RF_005,
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El funcionario elige Gestionar Solicitud	2. El sistema muestra la interfaz Gestionar Solicitud donde se pueden realizar las siguientes opciones: • Crear solicitud • Modificar solicitud • Eliminar solicitud • Enviar solicitud • Buscar solicitud • Mostrar detalles de la solicitud • Cambiar estado de la solicitud • Dar respuesta a la solicitud Las funcionalidades Modificar solicitud, Eliminar solicitud y Enviar solicitud son solo en caso que la

	solicitud este en estado de creada. La funcionalidad Cambiar estado de la solicitud y Dar respuesta a la solicitud estarán disponible solo para el Director de la DCV.
3. El funcionario elige la opción Salir. Si selecciona la opción Crear Solicitud ver la sección Crear Solicitud. Si selecciona la opción Modificar solicitud ver la sección Modificar solicitud. Si selecciona la opción Eliminar solicitud ver la sección Eliminar solicitud. Si selecciona la opción Enviar solicitud ver la sección Enviar solicitud. Si selecciona la opción Buscar solicitud ver la sección Buscar solicitud. Si selecciona la opción Mostrar detalles de la solicitud ver la sección Mostrar detalles. Si selecciona la opción Cambiar estado de la solicitud ver la sección Cambiar estado de la solicitud. Si selecciona la opción Dar respuesta a la solicitud ver la sección Dar respuesta a la solicitud.	4. El sistema cierra la interfaz. Se termina así el caso de uso.

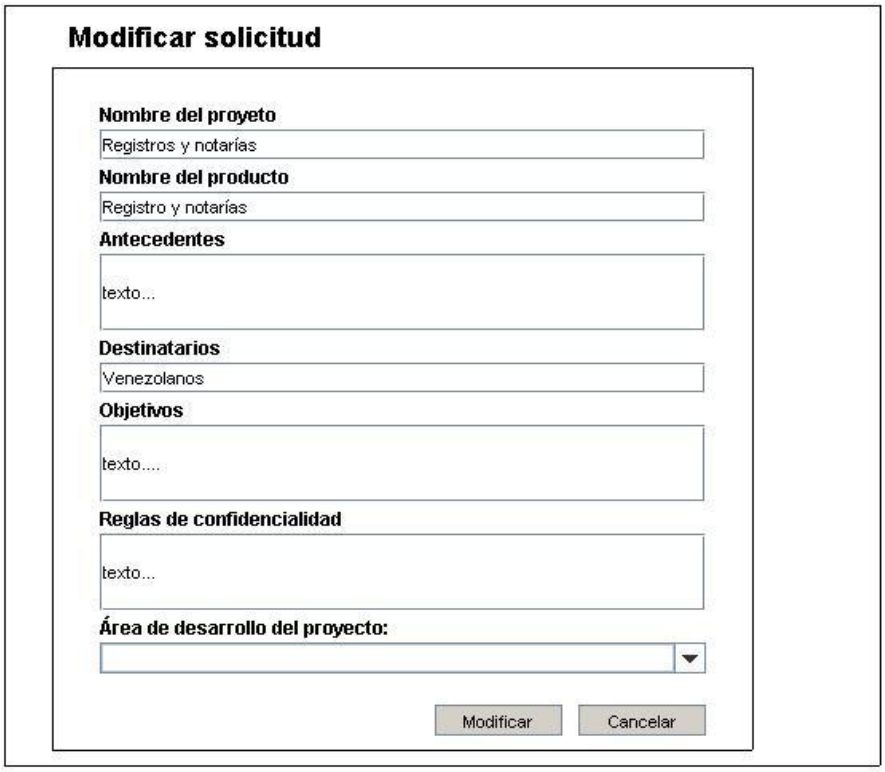


Sección	
Sección "Crear solicitud"	
Acción del actor	Respuesta del sistema
	1. El sistema muestra la interfaz con los siguientes datos de la solicitud: • Nombre del proyecto • Nombre del producto • Antecedentes • Destinatario • Objetivos • Reglas de confidencialidad • Área de desarrollo el proyecto
2. El solicitante elige crear. Si desea Enviar ver sección "Enviar	3. El sistema crea la solicitud. Terminando así el caso de uso.

solicitud". Si desea cancelar ver flujo alterno 1 "Cancelar"	

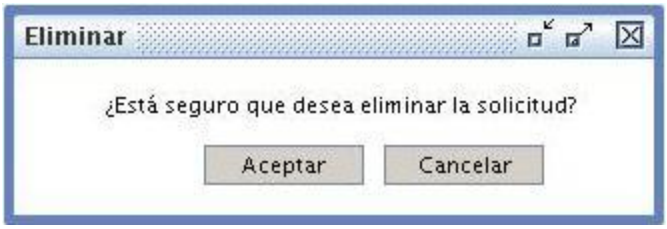
Sección "Modificar solicitud"	
Acción del actor	Respuesta del sistema
1. El Solicitante elige una solicitud. 2. El Solicitante elige la opción Modificar Solicitud.	3. El sistema muestra la interfaz modificar solicitud con los datos cargados. Se pueden modificar los siguientes datos: • Nombre del proyecto • Nombre del producto • Antecedentes

	• Destinatario • Objetivos • Reglas de confidencialidad • Área de desarrollo el proyecto
4. El solicitante introduce los datos. 5. El solicitante elige modificar. Si desea cancelar la operación ver flujo alterno 1 “Cancelar”.	6. El sistema modifica los datos. Termina así el caso de uso.



Sección “Eliminar solicitud”	
Acción del actor	Respuesta del sistema
1. El Solicitante elige una solicitud. El solicitante elige eliminar.	2. El sistema muestra un mensaje de confirmación.

3. El solicitante elige aceptar. Si desea cancelar ver flujo alterno 1 "Cancelar"	4. El sistema elimina la solicitud. Terminado así el caso de uso.



Sección "Enviar solicitud"	
Acción del actor	Respuesta del sistema
1. El Solicitante elige una solicitud que esté en estado de creada. 2. El Solicitante elige "Enviar".	3. El sistema muestra un mensaje de confirmación.
4. El solicitante elige aceptar. Si desea cancelar ver flujo alterno 1 "Cancelar"	5. El sistema envía la solicitud. Terminado así el caso de uso.

Sección "Buscar solicitud"	
Acción del actor	Respuesta del sistema
1. El solicitante introduce los datos de búsqueda. Si desea hacer una búsqueda avanzada ver el flujo alterno "Búsqueda avanzada".	2. El sistema hace una búsqueda por los criterios siguientes: • Proyecto • Producto • Fecha • Estado • Número Termina así el caso de uso.

Sección "Mostrar detalles de solicitud"	
Acción del actor	Respuesta del sistema
1. El solicitante elige una solicitud. El solicitante elige la opción "Detalles".	2. El sistema muestra los detalles de la Solicitud. Se muestra los siguientes datos: • Número de la solicitud • Nombre del proyecto • Nombre del producto • Antecedentes • Destinatario • Objetivos • Reglas de confidencialidad • Área de desarrollo el proyecto • Solicitante • Fecha • Estado
3. El solicitante elige "Salir"	4. El sistema cierra la interfaz. Termina así el caso de uso.

Sección "Cambiar estado de solicitud"	
Acción del actor	Respuesta del sistema
1. El solicitante decide cambiar el estado de solicitud de "Pendiente" a "Aceptada" o "Pendiente".	2. El sistema permite cambia el estado de la solicitud. Si cambia el estado a "Aceptada" el sistema muestra una interfaz para introducir los datos siguientes de la reunión de inicio: • Fecha • Lugar • Hora • Nota Si cambia el estado a "Rechazada" el sistema muestra una interfaz para introducir los datos del motivo de rechazo:

	• Motivo • Nota
3. El solicitante introduce los datos. Si el estado fue cambiado a “Aceptado” el solicitante elige Guardar. Si el estado fue cambiado a “Rechazado”, el solicitante elige Acepta. Si desea cancelar la operación ir al flujo alterno 1 “Cancelar”.	4. El sistema guarda los datos. Termina así el caso de uso.
Sección “Dar respuesta a la solicitud”	
Acción del actor	Respuesta del sistema
1. El solicitante elige una solicitud que esté “Aceptada” que no haya sido enviada. El solicitante elige Enviar.	2. El sistema muestra un mensaje de confirmación.
3. El solicitante elige aceptar la confirmación. Si decide cancelar ver el flujo alterno Cancelar.	4. El sistema envía la respuesta. Termina así el caso de uso.

Flujos Alternos	
Flujo Alterno 1 Cancelar	
Acción del Actor	Respuesta del Sistema
1. El solicitante elige cancelar	2. El sistema cancela la operación. Termina así el caso de uso.
Flujo Alterno 2 Búsqueda Avanzada	
Acción del Actor	Respuesta del Sistema
1. El solicitante elige búsqueda avanzada.	2. El sistema muestra la interfaz Búsqueda Avanzada con los criterios: criterios siguientes: • Proyecto • Producto • Fecha • Estado • Número
3. El solicitante introduce los datos.	5. El sistema hace una búsqueda por los criterios especificados. Termina así el caso de uso.

4. El solicitante elige buscar.	
Poscondiciones	Debe quedar registrada la solicitud hecha a la DCV.

El caso de uso **Gestionar Solicitud**, es uno de los más importantes para el cliente, ya que este tiene implícito las funcionalidades que desencadenan un flujo importante de procesos. Tan es así que sin este caso de uso, no sería viable realizar una herramienta de gestión de proyectos para esta entidad, pues en ese caso se podría utilizar cualquier otra herramienta de gestión de proyectos como el Redmine.

A través de este caso de uso cobran vida un grupo importante de procesos, los cuales llevan implícitos el trabajo de todo el equipo de diseño y calidad de la DCV.

Gestionar solicitud es uno de los casos de uso que marca la diferencia, pues debe trabajar con un flujo de procesos diferentes al de otras herramientas de gestión de proyectos tales como el Redmine y el Microsoft Project, de ahí su valor arquitectónico.

2.5 Análisis

Esta es una etapa muy importante en la creación de un sistema de software, ya que constituye una visión general del sistema que puede ser más difícil de obtener por medio de los resultados del diseño y de la implementación. Es además un primer acercamiento a todos los elementos que conforman e interactúan en el sistema. Estos elementos poseen además un alto grado de abstracción, lo cual facilita la asimilación por parte de los desarrolladores que implementarán el sistema, así como por el resto del equipo de trabajo.

2.5.1 Clases del Análisis

Las clases del análisis representan abstracciones de conceptos, en las cuales deben incluirse atributos y operaciones a un alto nivel, por lo que no se incluye el paso de parámetros ni el tipo de datos. Las clases del análisis presentan 3 estereotipos básicos que están estandarizados en UML y se utilizan para ayudar al equipo de desarrolladores, como son: las clases de interfaz, las clases controladoras y las clases de entidad, donde cada una de ellas tiene su propio símbolo y funcionalidad.

A continuación se muestra una visión del diagrama de clases del análisis modelado para el caso de uso **Gestionar Cronograma**

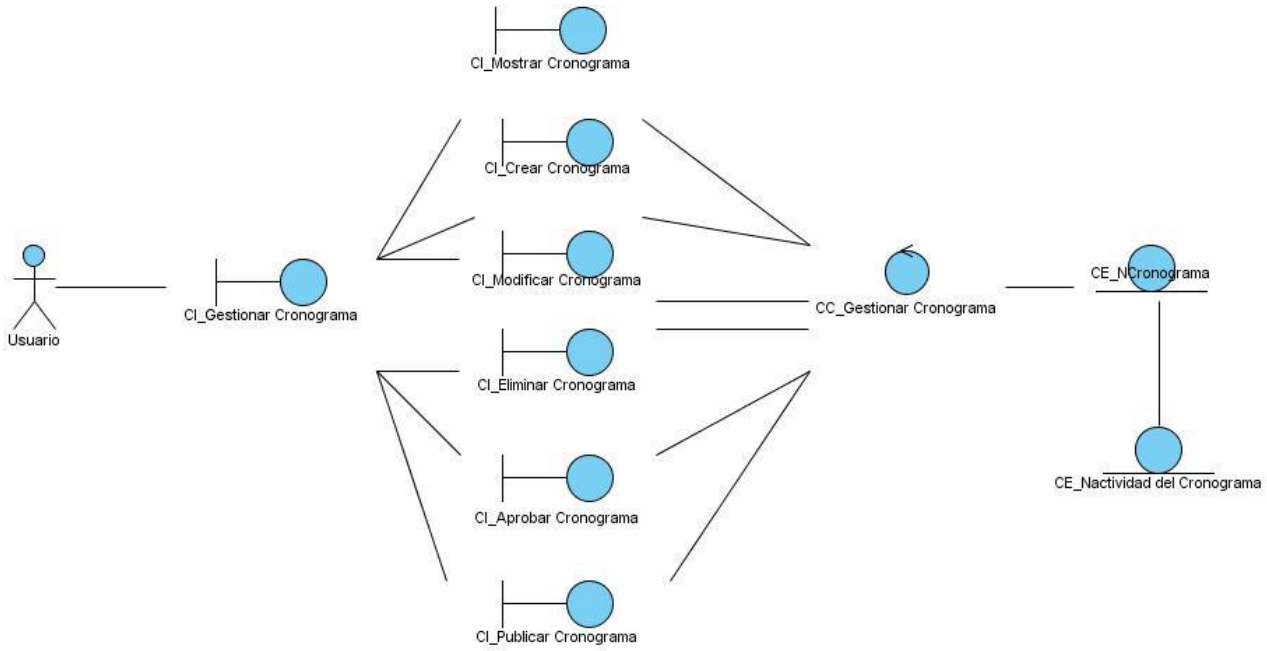


Figura 13 Diagrama de Clases del Análisis. CU Gestionar Cronograma

2.5.2 Colaboración en el Análisis

Con el objetivo de identificar requisitos y responsabilidades sobre los objetos es que se realizan los diagramas de colaboración. En estos se muestran las interacciones entre los objetos creando enlaces entre ellos y añadiendo mensajes a estos enlaces. Los mensajes denotan el propósito del objeto invocante en la interacción con el objeto invocado. (Jacobson, y otros, 2000)

A continuación se muestran los diagramas de colaboración en el análisis de algunos de los casos de uso más importantes del sistema.

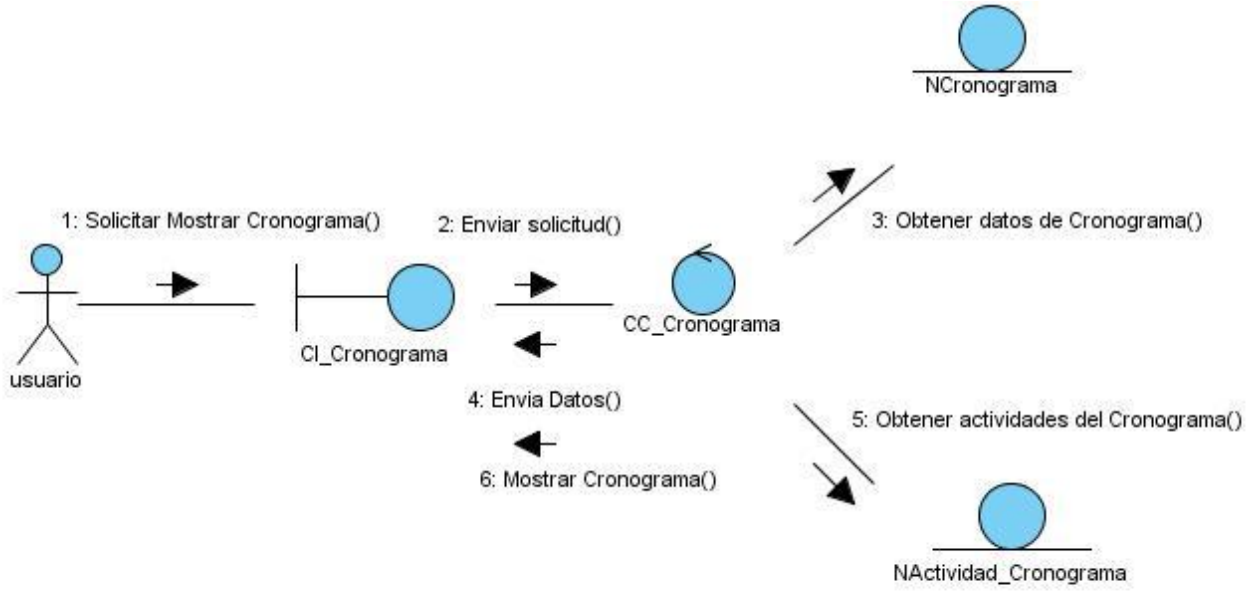


Figura 14 Diagrama de Colaboración en el Análisis: CU Gestionar Cronograma

2.6 Paquete de diseño

El diseño de paquetes constituye una división del sistema en partes más pequeñas y por consiguiente más manejables, estas no necesariamente tienen una interfaz definida. Facilita además la agrupación de clases con sus relaciones específicas, realizaciones de casos de uso, diagramas y otros paquetes relacionados de alguna manera. Se persigue el objetivo de lograr una mejor organización y especialización de los elementos del diseño, por lo que se definieron los siguientes paquetes para el Sistema para la Gestión de los Procesos de la DCV.

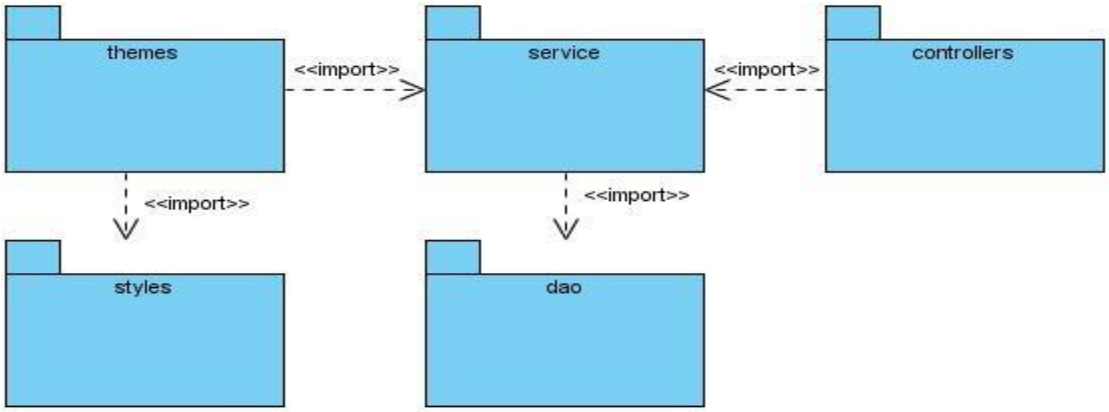


Figura 15 Paquetes de diseño del Sistema para la Gestión de los Procesos de la DCV.

2.7 Diagrama de clases del diseño

Los diagramas de clases del diseño representan las relaciones entre clases, interfaces y la colaboración entre estos elementos. Constituyen la vista del diseño estático del sistema. Estas clases ayudan a construir el sistema a través de la ingeniería directa e inversa de código. En ellas están contenidas las definiciones de las entidades de software. Se encargan de visualizar, estructurar y documentar los modelos.

El Sistema para la Gestión de los Procesos de la DCV, por razones arquitectónicas, se decidió hacerlo sobre el CMS Drupal. A continuación se muestran las clases del diseño tal y como funcionan modularmente en Drupal, con el objetivo de ilustrar la colaboración real y la representación de todos los elementos que interactúan en la ejecución de todas las funcionalidades que debe brindar el sistema.

Se muestra diagrama de clases de diseño del caso de uso **Gestionar Solicitud** del Sistema para la Gestión de los Procesos de la DCV. El resto de los diagramas de clases del diseño puede ser consultado en el documento referente al modelo del diseño habilitado a tales efectos.

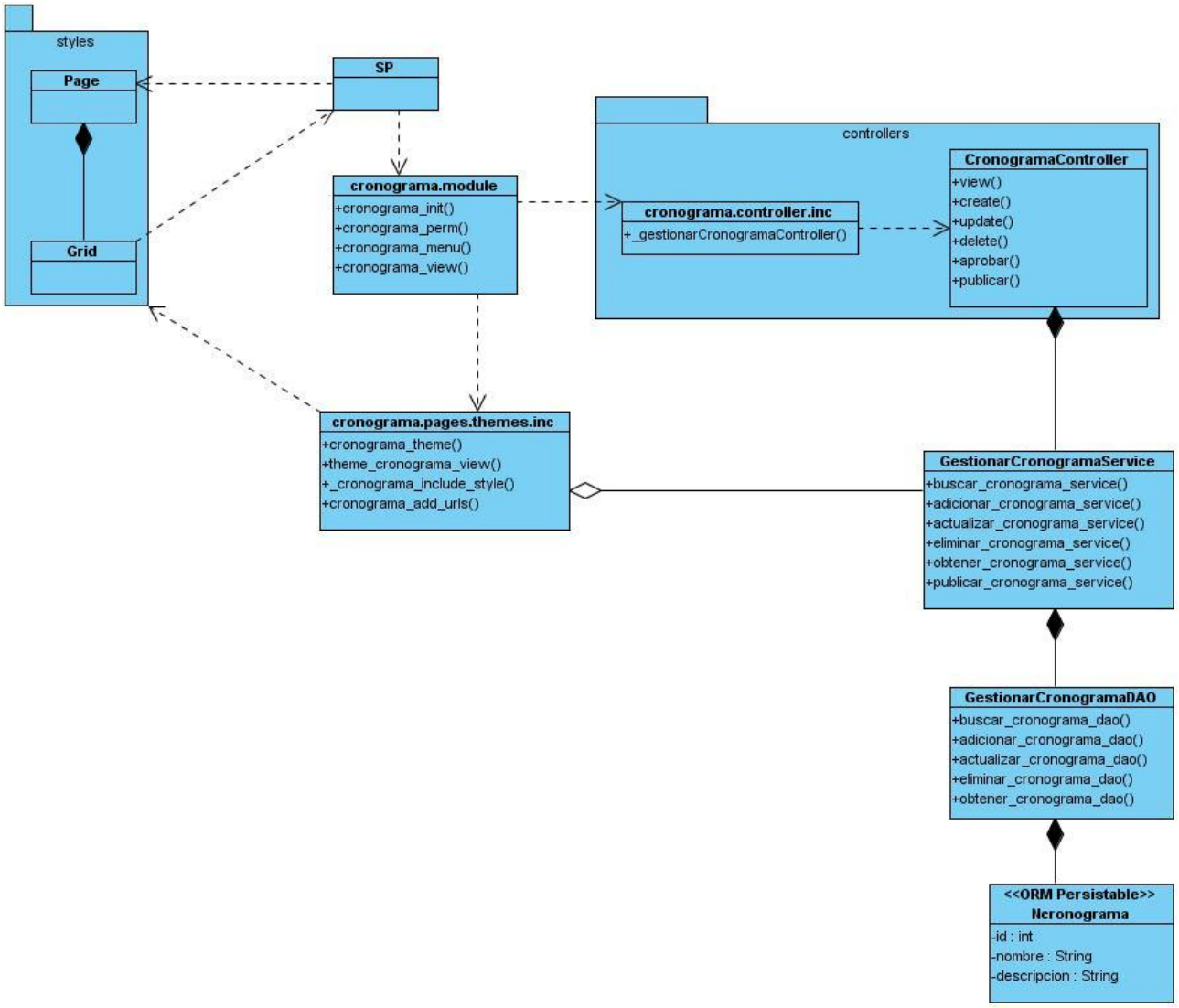


Figura 16 Diagrama de clases del diseño, CU Gestionar Cronograma.

2.8 Diagrama de secuencia

Los diagramas de secuencia muestran la interacción entre los diferentes objetos que están presentes en el sistema, esta secuencia esta ordenada de forma que se entienda correctamente todo el flujo de información que ocurre durante la llamada de una acción o método.

Se realiza un diagrama de secuencia para cada sección de los casos de uso para una mejor comprensión de los mismos.

A continuación se muestra un ejemplo de diagramas de secuencia de dos casos de uso muy importantes en el sistema, estos son **Gestionar Solicitud**, y **Gestionar Cronograma.**

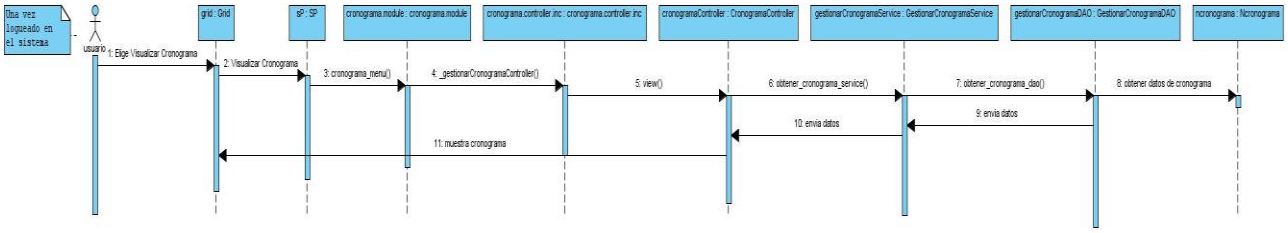


Figura 17 Diagrama de Secuencia Sección: Visualizar Cronograma, CU Gestionar Cronograma

2.9 Diagrama de clases persistentes

Con el objetivo de garantizar la persistencia de los datos de forma correcta se realizó el Diagrama de Clases Persistentes. Este diagrama representa la relación de los objetos persistentes con sus respectivas relaciones, dependencias y cardinalidades. A continuación se muestra una porción de este diagrama, la cual ilustra de forma general como quedó concebido el mismo.

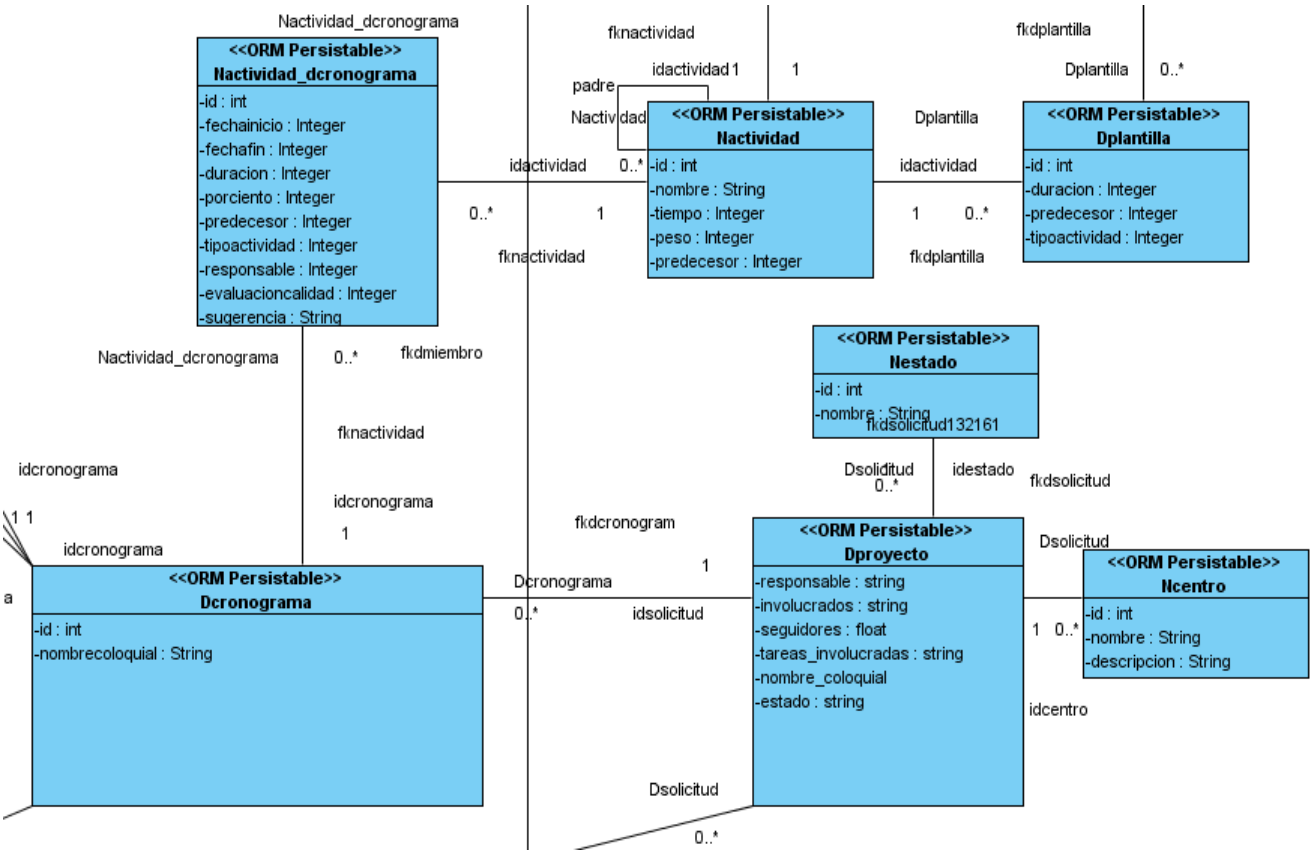


Figura 18 Diagrama de Clases Persistentes

2.10 Modelo de datos

El modelo de datos describe el comportamiento de los elementos físicos que persisten para dar soporte a la información que almacenará el sistema, con sus atributos correspondientes. Con el objetivo de garantizar la persistencia de los datos se modeló y normalizó el modelo de entidad relación del sistema para la gestión de los procesos de la Dirección de Comunicación Visual. De este se muestra seguidamente una porción, que demuestra de forma general como se garantiza la persistencia de datos.

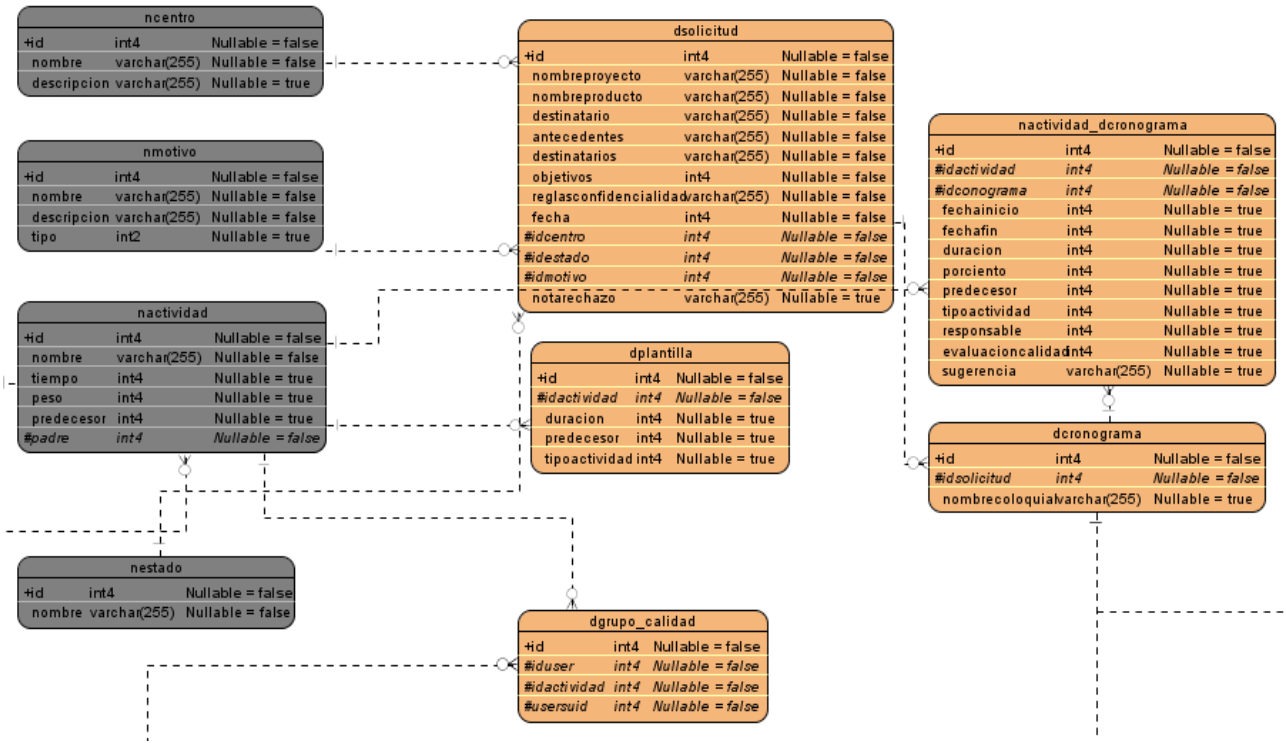


Figura 19 Modelo Entidad Relación

2.11 Conclusiones

En este capítulo se hace un resumen de los artefactos generados tanto en el análisis como en el diseño del sistema para la gestión de los procesos de la DCV, asumiendo las siguientes conclusiones:

- Se obtuvo el modelo de procesos de la DCV, siendo este un importante punto de apoyo para identificar los procesos que tienen lugar en esa entidad y que deben automatizarse.

- Partiendo del modelo de procesos y aplicando las técnicas de Ingeniería de Requisitos estudiadas en el Capítulo 1 se identificaron las funcionalidades y las características del sistema a desarrollar.
- Se generaron los artefactos necesarios para documentar los requisitos funcionales y no funcionales del sistema, así como la descripción de los Casos de Uso. Se identificaron además los actores que interactúan con el sistema, llegando a un acuerdo satisfactorio con el cliente en cuanto a funcionalidades y servicios que deberá prestar el sistema.
- Se construyeron los artefactos necesarios para dar soporte al modelo del diseño, lo que permitió obtener cada uno de los elementos que soportan los requerimientos funcionales y no funcionales del sistema, siendo la entrada a las actividades de implementación en un futuro.
- La aplicación de patrones de caso de uso, así como patrones de diseño permitieron obtener artefactos aceptables.

Capítulo 3. Análisis de los resultados

3.1 Introducción

Este capítulo tiene como objetivo mostrar los procedimientos y métodos empleados para medir la factibilidad de los artefactos obtenidos en el modelo del sistema así como en el modelo del diseño

Para la validación del modelo del sistema se usa el método de Kano (Kano, 1984), el cual fue escogido por el alto grado de certeza que brinda en cuanto en cuanto a satisfacción del cliente.

Para la validación de los artefactos del modelo del diseño se aplican métricas a nivel de componentes y del tamaño de clases.

Se debe garantizar la calidad de los artefactos obtenidos en esta etapa de análisis y diseño, pues es crucial para la posterior implementación del sistema. Esto ayuda sin dudas a minimizar los errores y obtener un producto de mayor calidad que cumpla con las expectativas del cliente, de ahí la importancia del análisis de los resultados obtenidos en cada etapa del proceso de desarrollo de software.

3.2 Validación del modelo del sistema

Los requerimientos identificados en las entrevistas con el cliente se especificaron en el artefacto llamado ¨Especificación de Requisitos¨, este artefacto junto al ¨Modelo del Sistema¨, son el producto de los análisis de los requerimientos e identificación de casos de uso, actores y descripción de ambos. Estos constituyen los principales artefactos generados durante el análisis del ¨Sistema para la gestión de los procesos de la DCV¨.

Con el objetivo de garantizar su factibilidad, los requerimientos elicitados fueron sometidos a varias revisiones con el cliente. Para constatar que estos cumplen con las expectativas del cliente, se aplica el método de Kano, por ser muy preciso en la relación satisfacción – funcionalidad que es precisamente lo que se desea lograr.

3.2.1 Selección de los requisitos y encuesta

Para lograr una heterogeneidad en los requisitos analizados se toma una muestra de la población en general, esta muestra de funcionalidades está compuesta por requisitos obligatorios de alta, media y baja complejidad, así como por requerimientos no funcionales que le resultan atractivos a los clientes, incrementando el valor añadido del sistema.

A continuación se muestran una parte de los requisitos analizados así como el formato en que se

realizó la encuesta a los clientes.

Requerimientos
Crear solicitud
Modificar solicitud (solo mientras esté en estado ¨creada¨)
Eliminar solicitud (solo mientras esté en estado ¨creada¨)
Enviar solicitud
Buscar solicitud
Mostrar detalles de la solicitud
Cambiar estado de la solicitud
Dar respuesta a la solicitud
Asignar proyecto
Reasignar un proyecto
Cambiar fase del proyecto(en curso o cerrado)
Visualizar cronograma
Adicionar actividades al cronograma
Modificar actividad del cronograma
Eliminar actividad del cronograma
Aprobar cronograma (cuando el jefe aprueba el cronograma hecho por el diseñador)
Publicar cronograma

Tabla 4 Requerimientos analizados para la validación

La encuesta fue realizada de forma presencial durante las entrevistas realizadas al cliente. Fueron entrevistados altos funcionarios de la Dirección de Comunicación Visual de la UCI, así como varios diseñadores y trabajadores de esa dirección.

Para la encuesta cada funcionario de la DCV debía responder a dos preguntas por cada requerimiento licitado, una pregunta medía la funcionalidad y la otra la disfuncionalidad del

requisito en cuestión. Para cada pregunta el encuestado tenía la posibilidad de seleccionar una de cinco respuestas posibles.

Funcionalidad	Pregunta	Respuesta
Crear solicitud	¿Cómo se sentiría si la aplicación contara con esta funcionalidad?	
	¿Cómo se sentiría si la aplicación NO contara con esta funcionalidad?	
Mostrar detalles de la solicitud	¿Cómo se sentiría si la aplicación contara con esta funcionalidad?	
	¿Cómo se sentiría si la aplicación NO contara con esta funcionalidad?	

Tabla 5 Formato de la encuesta aplicada

Opciones de Respuesta	
1	Me agrada
2	Es de esperarse
3	Neutral
4	Lo Acepto
5	Me Desagrada

Tabla 6 Opciones de respuesta a la encuesta

3.2.2 Representación de las respuestas

Las respuestas son agrupadas en una tabla de concentración, que se corresponde con cada una de las preguntas del cuestionario. El objetivo que se persigue con esta tabla es observar la dispersión de las respuestas.

Para la realización de la tabla que agrupa los datos de clasificación de cada uno de los requisitos según las respuestas a los encuestados, se utilizó la propuesta de León Duarte (León Duarte, 2005), como representación alterna a la clasificación de los requisitos. Esta interpretación de la clasificación (**Tabla 7**) de los requisitos se basa en el incremento de la satisfacción (Columna ¨Mejor¨), o en el decremento de la misma (Columna ¨Peor¨), debido a una inclusión o no de una característica o requerimiento del producto. Para la obtención de los valores ¨Mejor¨ y ¨Peor¨ se

utilizaron las formulas propuestas a continuación:

Mejor = (A+U) / (A+U+O+I) **Peor = (U+O) / (A+U+O+I)**

Requisitos Seleccionados para la Encuesta	A	O	U	I	Inv	D	Mejor	Peor	Total	C
Crear solicitud	2	4	1	1	1	1	0,37	0,62	10	O
Modificar solicitud (solo mientras esté en estado de creada)	1	5	1	1	1	1	0,25	0,75	10	O
Eliminar solicitud (solo mientras esté en estado de creada)	3	6		0	0	1	0,33	0,66	10	O
Enviar solicitud	1	8	0	0	0	1	0,11	0,88	10	O
Buscar solicitud	3	3	0	1	1	2	0,42	0,42	10	I
Mostrar detalles de la solicitud	6	2	1	1	0	0	0,7	0,3	10	A
Cambiar estado de la solicitud	4	3	1	0	1	1	0,62	0,5	10	U
Dar respuesta a la solicitud	3	5	1	1	0	0	0,4	0,6	10	O
Asignar proyecto	7	1	1	0		1	0,88	0,22	10	A
Reasignar un proyecto	3	5	1	1	0	0	0,4	0,6	10	O
Cambiar fase del proyecto(en curso o cerrado)	4	4	0	1	0	1	0,44	0,44	10	I
Visualizar cronograma	2	5	1	1	1	0	0,33	0,66	10	O
Adicionar actividades al cronograma	4	3	3	0	0	0	0,7	0,6	10	U
Modificar actividad del cronograma	3	4	1	0	0	2	0,5	0,625	10	U
Eliminar actividad del cronograma	5	3	0	0	0	2	0,625	0,375	10	A
Aprobar cronograma (cuando el jefe aprueba el cronograma hecho por el diseñador)	3	4	1	1	1	0	0,44	0,55	10	O
Publicar cronograma	3	4	1	1	0	1	0,44	0,55	10	O

Tabla 7 Respuesta de la encuesta

Clasificación	Abreviatura

Atractivo	A
Obligatorio	O
Unidimensional	U
Indiferencia	**I**

Tabla 8 Clasificación de los requisitos

Las fórmulas se obtienen de la percepción de ser Mejor que la competencia al satisfacer requisitos de tipo atractivos y unidimensionales, o bien, la de ser Peor que la competencia al no satisfacer requisitos del tipo unidimensional y obligatorios. En el denominador de ambas fórmulas aparece la suma de todos los atributos con excepción de las percepciones Inv. (Respuesta Inversa) y D. (Respuesta Dudosa), por su carácter confuso.

Estos valores obtenidos son graficados en una tabla que refleja de forma muy certera el grado de satisfacción del cliente.

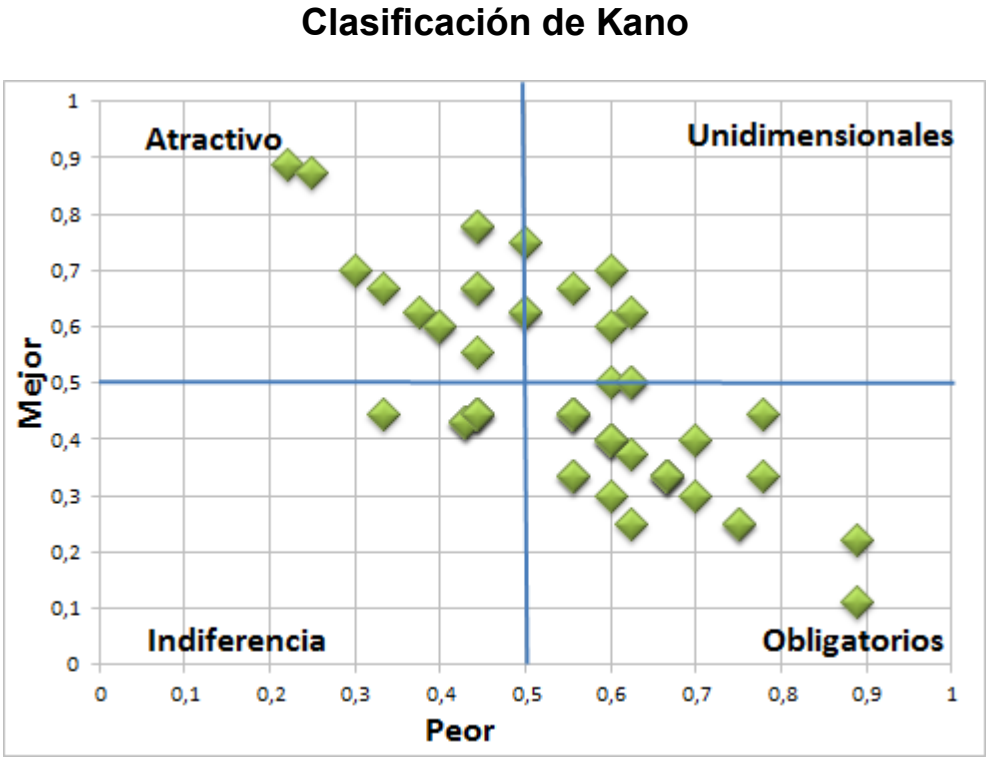


Figura 20 Clasificación de los requisitos en función de los resultados obtenidos

3.2.3 Clasificación de los Requisitos

Una vez analizados los datos recogidos en las encuestas se clasificaron los requisitos de la siguiente manera. De los 56 requisitos seleccionados para la encuesta, resultaron 12 ¨atractivos¨, 26 ¨obligatorios¨, 10 ¨unidimensionales¨ y 8 ¨indiferentes¨.

Los resultados evidencian que la satisfacción del cliente es elevada, ya que es buena señal que la mayoría de los requisitos se ubican en el rango de ¨obligatorios¨, lo cual infiere que la licitación de requisitos fue satisfactoria. En segundo lugar se ubican los requisitos ¨atractivos¨, 12, para ser exactos, esto demuestra un alto grado de valor añadido al sistema que se implementará, ya que resultan características que el cliente no se esperaba y resultan de su agrado al saber que podrá contar con ellas.

De igual forma se obtienen 10 requisitos ¨unidimensionales¨, lo cual evidencia mayores funcionalidades y mayor confort para el cliente.

Se obtienen además 8 funcionalidades como ¨indiferentes¨, estas pasan desapercibidas para el cliente, pero de igual forma se brindan en el sistema.

3.3 Métricas de diseño a nivel de componentes

3.3.1 Métrica de acoplamiento

Con el fin de medir el nivel de conectividad y dependencia del subsistema Cronograma con otros subsistemas se aplica la medida propuesta por Dhama (ver capítulo1). Se podrá medir además el nivel de dependencia y reutilización que posee el subsistema de acuerdo con el diseño propuesto.

Se define:

Acoplamiento de flujo de datos y de control:

d(i) = 2 número de parámetros de datos de entrada.

c(i) = 1 número de parámetros de control de entrada.

d(o) = 1 número de parámetros de datos de salida.

c(o) = 1 número de parámetros de control de salida.

Acoplamiento global:

g(d) = 0 número de variables globales usadas como datos

g(c) = 0 número de variables globales usadas como control.

Para el acoplamiento de entorno:

w = 3 número de subsistemas llamados (expansión).

r = 2 número de subsistemas que llaman al subsistema en cuestión (concentración).

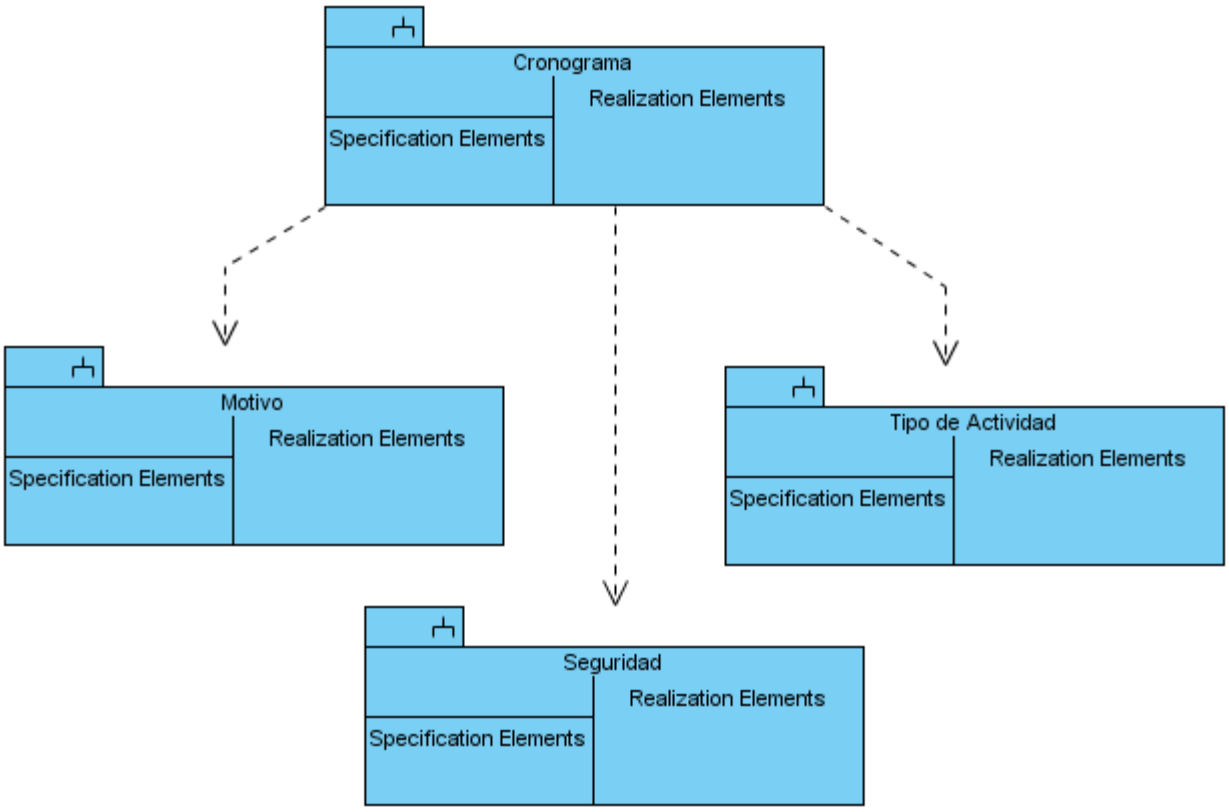


Figura 21 Expansión del Subsistema Cronograma

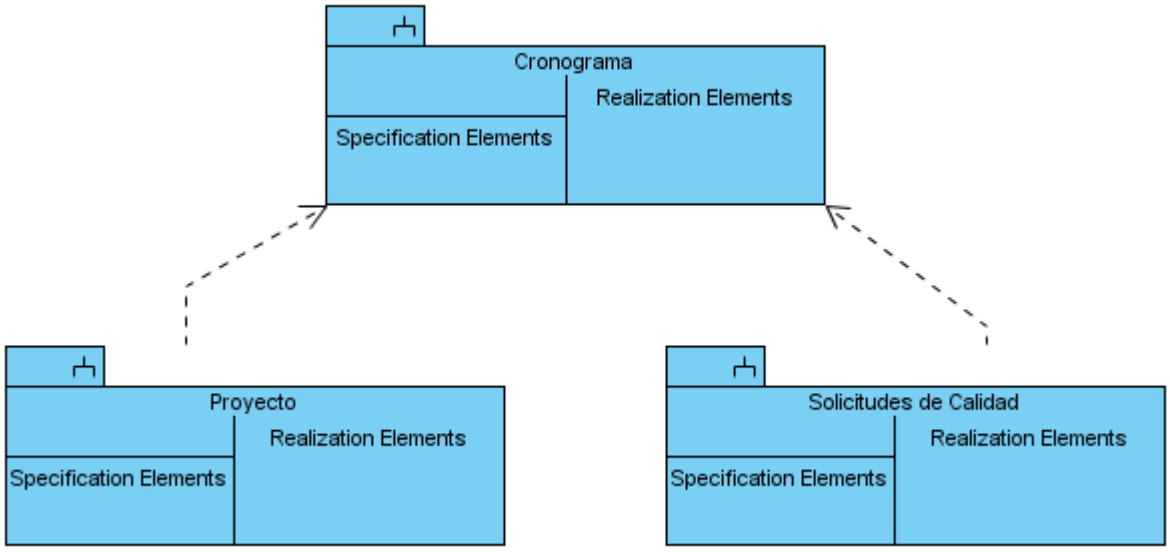


Figura 22 Concentración del Subsistema Cronograma

Indicador de acoplamiento del Subsistema:

$$m(c) = k/M$$

Donde K = 1, constante de proporcionalidad según el autor.

$$M = d(i) + a_x c(i) + d(o) + b_x c(o) + g(d) + c_x g(c) + w + r$$

a = b = c = 2.

m(c) = 1/ (2+ 2*1 +1+ 2*1 +0+2*0+3+2)

m(c) = 1/12

m(c) = 0.08

El valor del indicador de acoplamiento del subsistema Cronograma es de 0.08, esto indica el grado de acoplamiento del mismo, lo que sugiere que existe dependencia con otros subsistemas para su funcionamiento, sin embargo, se considera que esta es mínima, y no repercute por el hecho de depender de subsistemas básicos para el funcionamiento del sistema para la gestión de procesos de la DCV.

3.3.2 Métrica de Cohesión

Lograr una alta cohesión constituye una buena práctica de diseño, de ahí que se apliquen los patrones generales de software para asignación de responsabilidades GRASP(General

Responsibility Assignment Software Patterns) por sus siglas en inglés; en la construcción del mismo. Se aplican además las métricas de Bieman y Ott (ver capítulo 1) para medir el índice de cohesión de los elementos que conforman en diseño del subsistema Cronograma.

Para lograr esto se analizaron elementos fundamentales pertenecientes a los principales conceptos que plantea la métrica, como por ejemplo, Porción de datos, Símbolos Léxicos (tokens) de datos, Señales de súper-unión y Cohesión. Para analizar cada uno de estos conceptos se recogieron los datos que ilustran el nivel de uso de las principales clases diseñadas para el sistema de Gestión de los procesos de la DCV.

Clases	Usabilidad
GestionarCronogramaService	6
GestionarCargoService	1
GestionarSolicitudSErvice	1
CronogramaController	1
CargosController	1
SolicitudController	6
GestionarCronogramaDao	1
GestionarCargoDao	1
GestionarsolicitudDao	1
Cronograma.module	1
Cargo.module	0
Solicitud.module	0
NCronogramaORM	0
NcargoORM	1
DSolicitudORM	1

Tabla 9 Nivel de usabilidad de las clases

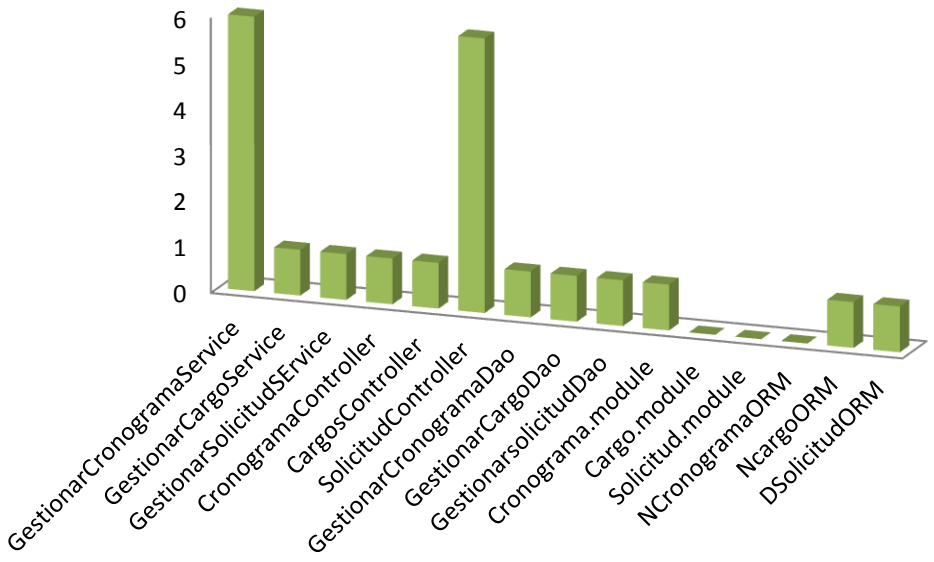


Figura 23 Gráfico de Usabilidad de las clases

La cohesión funcional se determina con dos enfoques:

1. Se determina la cohesión funcional fuerte (**CFF**) y la pegajosidad: se obtienen cuando el resultado de la métrica es de 1.

Se define como:

$$\boldsymbol{CFF} = \boldsymbol{número\ de\ súper\ adhesivos}\ (\boldsymbol{i}) / \boldsymbol{número\ de\ elementos}\ (\boldsymbol{i}).$$

2. Determinando la cohesión funcional débil (**CFD**):

Se define como:

$$\boldsymbol{CFD} = \boldsymbol{número\ de\ adhesivos}\ (\boldsymbol{i}) / \boldsymbol{número\ de\ elementos}\ (\boldsymbol{i}).$$

Adhesivo. Se le llamará adhesivo a un elemento que aparece en dos o más rebanadas.

Súper Adhesivo. Se denomina súper adhesivo a un elemento que está en todos los elementos de un módulo.

(i). Se define como la muestra.

Según los datos de las clases analizadas se tiene que:

número de elementos = 15

número de súper adhesivos(i) = 0

número de adhesivos(i) = 12

CFD = número de adhesivos (i) / número de elementos (i)

CFD = 12 / 15

CFD = 0.8

CFF = número de súper adhesivos (i) / número de elementos (i)

CFF = 0 / 15

CFF = 0

La métrica de Bieman y Ott plantea que mientras más cerca están los valores de CFF y CFD de 1 mayor será la cohesión del módulo. Los resultados demuestran que no hay una cohesión funcional fuerte, pero la relación del número de clases adhesivas con el número total de elementos de la muestra, determinados por la CFD = 0.8, está cercana a 1, lo cual demuestra que el diseño de las clases del Sistema para la gestión de los procesos de la DCV posee una cohesión funcional con un 80 % de fortaleza.

3.3.3 Métricas orientadas a clases. Tamaño de clase (TC)

Con el fin de comprobar el adecuado diseño de las clases y el nivel de reutilización de estas se aplicó la métrica del TC (ver capitulo 1).

Se aplicará esta métrica para las principales clases de 3 subsistemas definidos en el Sistema para la gestión de los procesos de la DCV.

Clases	Nro. de Atributos	Nro. de Operaciones
GestionarCronogramaService	1	6
GestionarCargoService	1	5
GestionarSolicitudSErvice	1	7
CronogramaController	6	6
CargosController	4	4

SolicitudController	8	8
GestionarCronogramaDao	0	24
GestionarCargoDao	0	14
GestionarsolicitudDao	0	22
Cronograma.module	0	4
Cargo.module	0	4
Solicitud.module	0	4
NCronogramaORM	3	0
NcargoORM	3	0
DSolicitudORM	10	0

Tabla 10 Cantidad de atributos y operaciones de las clases

Se presentó un **total de 15 clases** para un **promedio de atributos de 2.46** y un **promedio de operaciones de 4.33.**

De esta forma el umbral queda con los datos mostrados a continuación:

Umbral	Tamaño	Cantidad de clases
<= 20	Pequeño	13
>20 <=30	Medio	2
>30	Grande	0

Tabla 11 Clasicicación del tamaño de clases

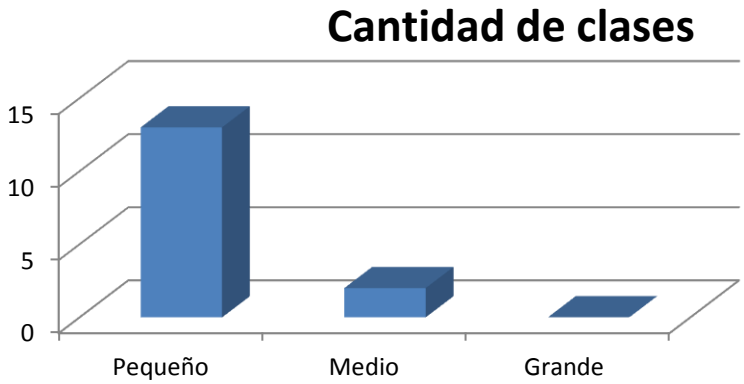


Figura 24 Gráfico de tamaño de clases

3.4 Conclusiones

En este capítulo se analizaron los resultados obtenidos durante el desarrollo del trabajo, se evaluaron además los artefactos que componen el modelo del sistema y los que componen el modelo del diseño, arribándose a las siguientes conclusiones:

- El método aplicado para la validación de los requerimientos elicitados demuestra que existe un alto grado de satisfacción del cliente, quedando bien especificados en el modelo del sistema, sin que se observen irregularidades en cuanto a ambigüedad de los mismos.
- El indicador de acoplamiento de los subsistemas del Sistema para la gestión de los procesos de la DCV determinó que solo existe dependencia minina con los módulos básicos del sistema, los cual no repercute en el correcto funcionamiento del sistema.
- El diseño de las clases del Sistema para la gestión de los procesos de la DCV posee una cohesión funcional con un 80 % de fortaleza, lo cual demuestra que sus elementos están altamente cohesionados.
- El 86% de las clases analizadas para la validación están consideradas como pequeñas, lo que facilitará sin dudas el proceso de construcción del Sistema para la gestión de los procesos de la DCV.
- Las métricas aplicadas a los elementos del diseño del Sistema para la gestión de los procesos de la DCV validan la calidad del diseño elaborado.

Conclusiones Generales

Al concluir el presente trabajo se arriba a las siguientes conclusiones:

- El análisis de los procesos de negocio, elaborado gracias a la interacción con los clientes del sistema y aplicando técnicas para la licitación de requisitos, permitió que se obtuvieran resultados satisfactorios en la identificación de los requerimientos que debe cumplir el sistema.
- La elaboración y especificación de los artefactos del modelo del sistema facilitó un mayor entendimiento y proporcionó un acuerdo común entre los clientes y el equipo de desarrollo, con respecto a las funcionalidades que debe brindar el sistema.
- Con la construcción de los artefactos del modelo del diseño, fue posible obtener los elementos que deberán ser implementados para obtener un sistema que cumpla con los requerimientos especificados por el equipo de análisis y aprobados por el cliente.
- El análisis de los artefactos obtenidos en el modelo del sistema y el de diseño permitió medir el grado de factibilidad de éstos en función de la calidad, funcionalidad, ambigüedad, reutilización y consistencia.

Recomendaciones

En aras de desarrollar el Sistema para la Gestión de los procesos de la Dirección de Comunicación Visual y posteriores mejoras el mismo, se recomienda:

- Desarrollar la implementación del Sistema para la Gestión de los procesos de la Dirección de Comunicación Visual.

Bibliografía

American Customer Satisfaction Index Homepage. National Quality Research Center. [Online] [Cited: abril 2, 2009.] http://www.theacsi.org/.

Burcio Sánchez, Mónica. 2009. *DEFINIDOR VISUAL BAJO ECLIPSE.* Madrid : s.n., 2009.

2011. Drupal. [Online] 2011. http://drupal.org/.

Durán Toro, Amador. 2000. *Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información.* Sevilla : s.n., 2000.

Escalona, José María and Korch, Nora. 2002. *Ingeniería de requisitos en Aplicaciones para la Web- Un estudio comparativo.* 2002.

Gido, Jack and Clements, James P. 1999. *Administración Exitosa de Proyectos.* s.l. : International Thomson Editores, 1999. 968-7529-84-9.

Humphrey, Watts S. 2001. *Introducción al Proceso de Software Personal.* Madrid : Pearson Educación, 2001. 84-7829-052-4.

Hurtado, Julio Ariel and Bastiarrica, Cecilia . 2005. *Modelo de Procesos, Calidad y Mejoramiento: CMM, TSP, PSP, ISO, IEEE, SPICE.* 2005.

IBM. 2006. *Rational Unified Process.* s.l. : IBM Corp., 2006.

IEEE. 1990. *IEEE Computer Dictionary-Compilation of IEEE Standard Computer Glossaries.* 1990.

Jacobson, Ivar, Booch, Grady and Rumbaugh, James. 2000. *El proceso unificado de desarrollo de software.* Madrid : Pearson Educación, 2000. 84-7829-036-2.

Kano, Noriaki. 1984. *Miryokuteki Hinshitsu to atarimae Hinshitsu (Calidad Atractiva y Calidad obligatoria).* Tokio : Japanese Society for Quality Control JSQC , 1984.

Knowledge Based Systems, Inc. 2010. Integrated Definition Methods. [Online] 2010. [Cited: febrero 22, 2011.] http://www.idef.com/.

Kruchten, P. 2000. *The Rational Unified Process: An Introduction .* s.l. : Addison Wesley, 2000.

Larman, Craig. 1999. *UML y Patrones. Introducción al análisis y diseño orientado a ojeto.* México : Adisson Wesley, 1999. 970-17-0261-1.

BIBLIOGRAFÍA

León Duarte, Jaime alfonso. 2005. *Metodología para la detección de requisitos subjetivos en el diseño del producto.* La Habana : s.n., 2005.

Martínez, Rafael. 2011. PostgreSQL. [Online] junio 8, 2011. http://www.postgresql.org.es/sobre_postgresql.

Microsoft. 2007. *Microsoft Office Project Help.* s.l. : Microsoft, 2007.

Philippe Lang, Jean. 2006-2011. Redmine. [Online] 2006-2011. [Cited: febrero 11, 2011.] http://www.redmine.org/guide.

—. 2006-2011. Redmine. [Online] 2006-2011. [Cited: febrero 11, 2011.] http://portal.dt.prod.uci.cu/projects/ayuda/wiki.

Pressman, Roger S. 2001. *Ingeniería del Software. Un enfoque práctico.* Madrid y Carchelejo : s.n., 2001.

Rumbaugh, James, Jacobson, Ivar and Booch, Grady. 1998. *El lenguaje Unificado de Modelado. Manual de Referencia.* 1998.

Sommerville, Ian. 2005. *Ingeniería de Software.* Madrid : Pearson Educación, 2005.

2010. Sparx Systems. [Online] Sparx Systems Pty Ltd., 2010. [Cited: febrero 22, 2011.] http://www.sparxsystems.com.ar/products/ea/index.html.

Sparx Systems Pty. Ltd. Sparx Systems. [Online] [Cited: junio 3, 2011.] http://www.sparxsystems.com/products/ea/index.html.

Una introducción al UML. El Modelo de Casos de Uso. **Sparks, Geoffrey.** Australia : s.n.

Use Cases: Patterns and Blueprints. **Overgaard, Gunnar and Palmkvist, Karin. 2004.** s.l. : Addison Wesley, 2004.

Visual Paradigm. [Online] [Cited: febrero 23, 2011.] http://www.visual-paradigm.com/product/vpuml/.

Wake, William C. 2002. *Extreme Programing Explored.* s.l. : Addison-Wesley, 2002.

World Wide Web Consortium. 1994. W3C World Wide Web Consortium. [Online] 1994. http://www.w3c.org..

GLOSARIO DE TÉRMINOS

Glosario de términos

UCI: Universidad de las Ciencias Informáticas.

DCV: Dirección de Comunicación Visual de la Universidad de Ciencias Informáticas.

Actores: Roles pertenecientes a los usuarios, agrupados según sus iteraciones con las funcionalidades del sistema.

CASE: (Computer Aided Software Engineering), Ingeniería de Software Asistida por Ordenador).

Caso de uso (CU): Representación de la agrupación de funcionalidades comunes. Representan un conjunto de iteraciones entre el sistema y sus actores.

CRUD: Patrón de casos de uso. Plantea que las funcionalidades de crear, obtener, actualizar y eliminar (Create, Read, Update y Delete), se pueden agrupar en un mismo CU.

GRASP (General Responsibility Assignment Software Patterns): Patrones generales de software para asignación de responsabilidades.

Paquetes de diseño: Es una colección de clases, relaciones, realizaciones de casos de uso, diagramas y otros paquetes que estén de alguna forma relacionados. No definen un comportamiento o semántica bien definida.

Requerimientos: Condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.

Subsistemas de diseño: Representan una separación de aspectos lógicos de diseño. Constituyen una forma de organizar los artefactos del modelo de diseño en piezas más manejables. Puede contener clases del diseño, realizaciones de casos de uso, interfaces y otros subsistemas.

RUP (Rational Unified Process): Proceso Unificado se software (Metodología de desarrollo de software).