

Universidad de las Ciencias Informáticas

Facultad 3



**Análisis y diseño del módulo Gestión de Trámites para
la Salida al Exterior del Sistema Integral de Cooperación
Internacional de la Facultad 3.**

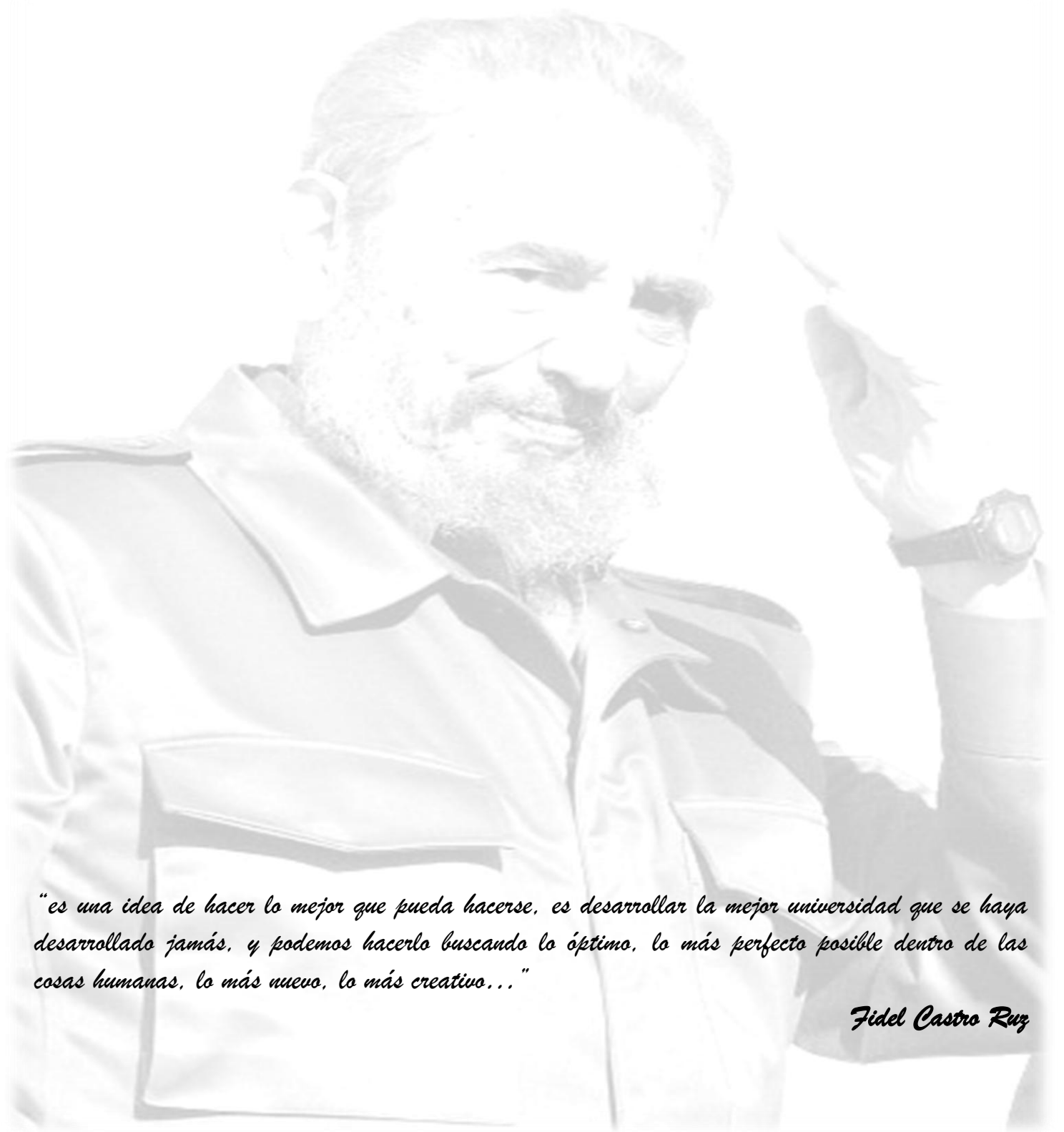
Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autor: Aracelys Muñoz Marín

Tutor: Ing. Ariagna González Landeiro

Ciudad de La Habana

2011



"es una idea de hacer lo mejor que pueda hacerse, es desarrollar la mejor universidad que se haya desarrollado jamás, y podemos hacerlo buscando lo óptimo, lo más perfecto posible dentro de las cosas humanas, lo más nuevo, lo más creativo..."

Fidel Castro Ruz

DECLARACIÓN DE AUTORÍA

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

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

Aracelys Muñoz Marín

Ing. Ariagna González Landeiro

AGRADECIMIENTOS

A todos los que colaboraron en la realización del presente trabajo...

A todas las personas que durante estos 5 años aportaron a mi formación profesional y personal... A mi tutora por su ayuda y el tiempo dedicado...

A mis amigas (os) y compañeras (os), esos con los que compartí momentos inolvidables en la universidad, que estuvieron presente cuando más los necesitaba y que se ganaron mi cariño, en especial a Gretter y Dane por estar siempre para todo, a Damián todo lo que me inculcó en muchos años...

A mi familia, por el apoyo incondicional durante estos 5 años...

A mi tutora, por su ayuda y el tiempo dedicado durante el trabajo...

A Isabel, por las energías y atenciones que me fueron ofrecidas, pero sobre todo por la belleza de su persona...

A Yoe, por su amor, ternura, cariño, pasión y sobre todo por ser quien es...

A mi tía del alma, mis hermanos y queridos abuelos...

A Fidel, por su obra y confianza infinita en la universidad del futuro...

A la Revolución...

DEDICATORIA

A mis padres...

RESUMEN

Con el objetivo de incrementar el impacto de la Universidad de las Ciencias Informáticas en la informatización del país, diversificar el mercado, incrementar las exportaciones y fortalecer la formación pedagógica y científica del claustro de egresados y profesores, trabajadores y estudiantes tienen lugar en el centro un grupo de misiones de cooperación internacional.

En la Facultad 3 se realizan varios procesos, entre ellos se encuentra la Gestión de Trámites para la Salida al Exterior. Este proceso tiene como objetivo gestionar los trámites referentes a las solicitudes de misiones de Cooperación Internacional que se realizan desde las diferentes áreas de la facultad. Actualmente no se cuenta con una solución informática que permita realizar las actividades necesarias para la gestión de trámites de salida al exterior una vez que se solicita una misión por lo que las mismas, en este nivel, se continúan realizando sin aprovechar al máximo las tecnologías existentes en el centro y se ven afectadas por un grupo de deficiencias que ponen en riesgo su realización con la calidad requerida. De ahí la necesidad de emplear nuevas variantes que agilicen el flujo de actividades para tramitar la salida al exterior y que a su vez garanticen la calidad, eficiencia y rapidez de las mismas.

Por tanto, el objetivo principal del presente trabajo se centra en la realización del Análisis y Diseño de una aplicación web que gestione adecuadamente las actividades necesarias para tramitar una solicitud de misión de Cooperación Internacional con vistas a contribuir a la construcción de un sistema que mejore la eficiencia del proceso Gestión de Trámites para la Salida al Exterior.

El presente documento recoge un estudio sobre otros sistemas de trámites existentes en el ámbito nacional y en la Universidad, además se describen las características de las herramientas usadas para el desarrollo de la propuesta de solución. Se especifican de forma detallada las funcionalidades que debe brindar el sistema a sus clientes, propone además un diseño basado en el uso de un Sistema de Gestión de Contenido que facilite la implementación de la aplicación web propuesta.

Palabras Claves

trámites, misiones de Cooperación Internacional, procesos, aplicación web.

TABLA DE CONTENIDOS

AGRADECIMIENTOSI

DEDICATORIAII

INTRODUCCIÓN1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA7

 1.1 Introducción7

 1.2 Marco conceptual.....7

 1.3 Sistemas de Trámites Migratorios9

 1.3.1 Sistemas de Trámites Migratorios en Cuba9

 1.3.2 Valoraciones de los sistemas estudiados10

 1.4 Aplicaciones Web.....11

 1.5 Metodología de desarrollo12

 1.5.1 Principales metodologías de desarrollo13

 1.5.2 Valoración de las metodologías estudiadas14

 1.6 Modelo de desarrollo adoptado.....15

 1.7 Lenguajes de Modelado16

 Notación para el Modelado de Procesos del Negocio.....16

 Lenguaje de Modelado Unificado16

 1.7.1 Comparación entre UML y BPMN17

 1.8 Herramientas de Modelado17

 1.8.1 Visual Paradigm for UML 6.1 Enterprise Edition18

 1.9 Sistema Gestor de Contenido18

 1.10 Sistema Gestor de Base de Datos20

 1.11 Servidor Web20

1.12 Lenguajes de programación	21
1.12.1 Lenguajes de programación del cliente.....	21
1.12.2 Lenguajes de programación del servidor	22
1.13 Entorno Integrado de Desarrollo.....	22
1.14 Marcos de Trabajo	23
1.15 Conclusiones.....	23
CAPÍTULO 2: NEGOCIO Y ANÁLISIS DEL SISTEMA.....	24
2.1 Introducción	24
2.2 Modelamiento del Negocio	24
2.3 Procesos de Negocio	24
2.4 Descripción de Procesos de Negocio.....	26
2.4.1 Descripción del proceso Presentar propuesta de solicitud de misión.....	26
2.4.2 Descripción del proceso Crear planilla de solicitud de misión	28
2.4.3 Descripción del subproceso Aprobar solicitud de misión.....	29
2.5 Validación de los procesos de negocio	30
2.6 Modelo Conceptual	30
2.7 Requisitos	31
2.8 Técnicas de Captura de Requisitos.....	32
2.9 Identificación de requisitos	33
2.9.1 Requisitos Funcionales.....	33
2.9.2 Requisitos No Funcionales	37
2.10 Especificación de requisitos	39
2.11 Técnicas de validación de requisitos	39
2.12 Métricas de calidad de especificación de requisitos.....	41
2.13 Conclusiones.....	41

CAPÍTULO 3: DISEÑO DEL SISTEMA	42
3.1 Introducción	42
3.2 Arquitectura base	42
3.3 Patrón arquitectónico Modelo Vista Controlador.....	42
3.3.1 El patrón MVC en los componentes Joomla	44
3.4 Descripción y presentación del diagrama de componentes	44
3.6 Diagramas de Clases del Diseño	46
3.6.1 Descripción del diagrama de clases del diseño propuesto	47
3.7 Patrones de diseño	48
3.8 Conclusiones	50
CAPÍTULO 4: VALIDACIÓN DEL SISTEMA	51
4.1 Introducción	51
4.2 Aplicación de las técnicas para la validación de requisitos	51
4.2.1 Revisión Técnica Formal	51
4.3.2 Prototipos	51
4.3 Aplicación de las métricas de calidad de la especificación de requisitos	61
4.3.1 Resultados de las métricas aplicadas.	61
4.4 Evaluación del Modelo de Diseño	62
4.5 Resultados de la evaluación de las métricas empleadas.....	62
CONCLUSIONES	66
RECOMENDACIONES	67
REFERENCIAS BIBLIOGRAFICAS	68

ÌNDICE DE FIGURAS

FIGURA 1 ARQUITECTURA CLIENTE-SERVIDOR	12
FIGURA 2 GESTIONAR TRÁMITES DE SALIDA AL EXTERIOR.....	26
FIGURA 3 DIAGRAMA DEL PROCESO PRESENTAR PROPUESTA DE SOLICITUD DE MISIÓN	27
FIGURA 4 DESCRIPCIÓN DEL PROCESO CREAR PLANILLA DE SOLICITUD DE MISIÓN.....	28
FIGURA 5 DESCRIPCIÓN DEL SUBPROCESO APROBAR SOLICITUD DE MISIÓN.....	29
FIGURA 6 MODELO CONCEPTUAL	31
FIGURA 7 DIAGRAMA DE COMPONENTES PROPUESTO	45
FIGURA 8 MODELO DE DATOS PROPUESTO	46
FIGURA 9 DIAGRAMA DE CLASES DEL DISEÑO PROPUESTO	47
FIGURA 10 IU_ENVIAR SOLICITUD INTERNA.....	52
FIGURA 11 IU_LLENAR SOLICITUD	53
FIGURA 12 IU_GESTIONAR SOLICITUD	54
FIGURA 13 IU_SOLICITUD.....	55
FIGURA 14 IU_ACEPTAR VOTACIÓN.....	56
FIGURA 15 IU_ANALIZAR ESTADO DE LA VOTACIÓN	57
FIGURA 16 IU_REALIZAR VOTACIÓN	58
FIGURA 17 IU_ENTES QUE VETAN.....	59
FIGURA 18 IU_VINCULAR NOMENCLADOR MUNICIPIO SEGÚN PROVINCIA.....	59
FIGURA 19 IU_CONFIGURAR NOMENCLADORES.....	60
FIGURA 20 IU_CREAR NOMENCLADOR DINÁMICO.....	60
FIGURA 21 GRÁFICO CANTIDAD DE PROCEDIMIENTOS POR CLASES.....	63
FIGURA 22 GRÁFICO DE LOS RESULTADOS GENERALES DE ACUERDO A LA CANTIDAD DE PROCEDIMIENTOS	63
FIGURA 23 GRÁFICO DE DEPENDENCIAS ENTRE LAS CLASES.....	64
FIGURA 24 GRÁFICO DE ACOPLAMIENTO ENTRE CLASES	64

ÌNDICE DE TABLAS

TABLA 1 DESCRIPCIÓN DEL DISEÑO DE CLASES REALIZADO.....	47
TABLA 2 TABLA AUXILIAR PARA EVALUAR LA CALIDAD DE LA ESPECIFICACIÓN DE REQUISITOS.....	61
TABLA 3 APLICACIÓN DE LAS MÉTRICAS DE CALIDAD DE ESPECIFICACIÓN DE REQUISITOS.....	61

INTRODUCCIÓN

Con la introducción de las primeras computadoras y la preparación de nuevos profesionales desde la década de los 60 se inserta Cuba en el desarrollo de las TIC¹. Desde entonces se han venido diseñando y aplicado estrategias que permiten transformar los conocimientos y las tecnologías de la información y las comunicaciones en instrumentos a disposición del desarrollo de la informatización de la sociedad cubana, logrando la cultura digital como una de las características imprescindibles del hombre nuevo.

Conjuntamente con el desarrollo de las TIC se abre paso la industria del software en el país y comienzan a desarrollarse los primeros programas de informatización de la sociedad cubana. Actualmente esta industria está llamada a convertirse en una significativa fuente de ingresos para la nación a pesar de ser relativamente nueva y de verse frenada por las consecuencias del bloqueo económico, comercial y financiero impuesto por Estados Unidos. Inconvenientemente de los contratiempos externos son muchos los resultados alcanzados, los cuales han posibilitado la inserción de Cuba en el mercado mundial del software.

Las universidades cubanas han tenido un papel protagónico en la introducción y desarrollo en el país de las tecnologías computacionales, evidenciándose en la cantidad de trabajos investigativos y productivos obtenidos en las disciplinas relacionadas con las TIC y en los programas de formación de profesionales en los campos de la informática, electrónica y telecomunicaciones. Ejemplo de ello la Universidad de las Ciencias Informáticas (UCI), centro encargado de la formación y superación de profesionales de Informática así como de la producción de software para la industria nacional y la exportación.

La producción de Software y Servicios Informáticos en la UCI se basa en la integración de los procesos de formación, investigación y producción en torno a una temática para convertirla en una rama productiva. La misma se concentra en el desarrollo de proyectos que destacan resultados en las esferas de Salud, Educación, Software Libre, Teleformación, Sistemas Legales, Realidad Virtual, Automatización, Bioinformática, Procesamiento de imágenes y señales entre otros. Este espacio de integración contribuye a la preparación de fuerza de trabajo altamente calificada para el desarrollo de software y promueve

¹ Tecnologías de la Información y la Comunicación. Conjunto de tecnologías desarrolladas para gestionar información y enviarla de un lugar a otro.

además la formación pregrado, postgrado, el fomento de líneas de investigación y desarrollo y la colaboración nacional e internacional.

Vinculado a ello y con el objetivo de incrementar el impacto de la universidad en la informatización del país, diversificar el mercado, incrementar las exportaciones y fortalecer la formación pedagógica y científica del claustro de egresados y profesores, trabajadores y estudiantes tienen lugar en el centro un grupo de misiones de cooperación internacional. Las mismas se realizan por diferentes motivos, ya sean para comercializar, explorar, promover, comprar o vender productos y/o soluciones informáticas; intercambio con instituciones extranjeras; asistencia a seminarios, conferencias, congresos en el exterior o también para la realización de cursos, diplomados, maestrías, doctorados, entre otros.

La gestión de trámites de salida al exterior se distingue a partir del tipo de misión que se solicita, dependiendo de ésta comienzan a efectuarse una serie de pasos para tramitar la aprobación de la misma y la salida del solicitante una vez que es aprobada en los niveles correspondientes. Este proceso comprende el tiempo transcurrido desde que se realiza la solicitud de la misión hasta la fecha en que culmina la misma.

Una misión puede ser solicitada por un profesor investigador, especialista, directivo u otra persona a sus jefes inmediatos, la misma es presentada por los jefes de áreas para iniciar el trámite de aprobación de ésta. Le precede a ello otras actividades vinculadas al proceso de salida al exterior como el análisis de la solicitud por parte del área solicitante con los representantes políticos- sociales (PCC, UJC, FEU, CTC) y se somete la aprobación de la misión a nivel de facultad en el Consejo de Dirección. Una vez realizado esto es necesario llenar una Planilla de Salida al Exterior además de una serie de modelos establecidos en este proceso. Seguidamente la misión es sometida a Rectoría y de ser aprobada, la Oficina de Trámites se encarga de tramitar una serie de documentos necesarios para la salida al exterior de cada una de las personas asociadas a las misiones aprobadas.

Actualmente la Facultad 3 no cuenta con una solución informática que permita realizar las actividades necesarias para la gestión de trámites de salida al exterior una vez que se solicita una misión por lo que las mismas se continúan realizando sin aprovechar al máximo las tecnologías existentes en el centro y se ven afectadas por un grupo de deficiencias que ponen en riesgo su realización con la calidad requerida. Existen grandes posibilidades de que no se cumpla con el tiempo establecido para efectuar cada una de las actividades referentes al trámite de salida al exterior pues para cada tipo de misión se realizan distintas

actividades y las mismas son tramitadas por canales diferentes. Por ende son muchos los individuos que intervienen en este proceso, lo cual trae consigo que se realicen largos debates para concretar la aprobación de una misión ya que la información y documentación que se maneja es abundante. Sumado a ello, los solicitantes, en su mayoría, no entregan las planillas de salida al exterior con todos los datos correspondientes por lo que se pierde tiempo completando las mismas y no existe la posibilidad de reutilizar los datos que son comunes entre ellas. Además no se cuenta con un registro detallado del resultado de la misión y el aporte que ofrece la misma. De ahí la necesidad de emplear nuevas variantes que agilicen el flujo de actividades para tramitar la salida al exterior y que a su vez garanticen la calidad, eficiencia y rapidez de las mismas.

Dada la **situación problemática** existente se plantea el siguiente **problema a resolver**:

El proceso existente en la Facultad 3 para gestionar los trámites de salida al exterior no garantiza agilidad en el mismo, ni una conservación histórica de los resultados de la misión.

Será **objeto de estudio** la Ingeniería de Software enmarcada en los procesos de Cooperación Internacional en la UCI y el **campo de acción** se centra en el proceso Gestión de Trámites de Salida al Exterior en la Facultad 3.

Objetivo General:

Realizar el análisis y diseño del módulo Gestión de Trámites para la Salida al Exterior del Sistema Integral de Cooperación Internacional de la Facultad 3.

Objetivos Específicos:

- Desarrollar el marco teórico de la investigación.
- Realizar la modelación de los procesos de negocio asociados a la gestión de trámites de salida al exterior en la Facultad 3.
- Realizar el análisis y el diseño del sistema.
- Validar el resultado obtenido.

Como **idea a defender** se plantea lo siguiente: Realizando el análisis y diseño del proceso Gestión de Trámites de Salida al Exterior en la Facultad 3 de la Universidad de las Ciencias Informáticas, se contribuirá al desarrollo de un sistema que garantizará agilidad en los procesos de trámites y una conservación histórica de los resultados de la misión.

Para dar cumplimiento a los objetivos propuestos se trazaron las siguientes **tareas investigativas**:

- Análisis de las misiones de cooperación internacional en la Universidad y del proceso de Gestión de Trámites de Salida al Exterior en la Facultad 3.
- Descripción de los principales conceptos y definiciones que se tratan durante el desarrollo de la investigación.
- Estudio de los sistemas existentes para la gestión de procesos de cooperación internacional a fin de identificar las principales características a tener en cuenta para el desarrollo de la solución.
- Justificación del uso de las tecnologías, lenguajes y herramientas propuestas para el desarrollo del sistema.
- Identificación de los procesos asociados a la Gestión de Trámites de Salida al Exterior en la Facultad 3.
- Realización de la captura y descripción de los requerimientos asociados a la Gestión de Trámites de Salida al Exterior.
- Realización del diseño del sistema.
- Validación de los resultados obtenidos.

Para el cumplimiento de estas tareas investigativas se emplearon los siguientes métodos de investigación:

Métodos Teóricos:

- **Histórico – Lógico:** Para realizar un estudio de las tendencias históricas y actuales de los sistemas referentes a la cooperación internacional, la metodología de desarrollo, herramientas y lenguajes a utilizar en la solución.

- **Analítico - Sintético:** Para realizar un análisis de la documentación e información a emplear para el desarrollo de la investigación
- **Modelación:** Para modelar el negocio y lograr un mejor entendimiento del problema que se está investigando y los procesos implicados en el mismo.

Métodos Empíricos:

- **Entrevistas:** Para realizar el estudio de los procesos actuales referentes a la Gestión de Trámites de Salida al Exterior y las particularidades de los mismos a fin de determinar los problemas existentes en la Facultad 3.

Para una mejor comprensión de la investigación, el trabajo se estructuró de la siguiente forma:

Capítulo 1: Fundamentación Teórica.

Se describen conceptos utilizados a lo largo del documento, facilitando así la comprensión de temas posteriores. Se realiza un breve estudio de otros sistemas existentes en Cuba similares al que se pretende modelar a fin de identificar las principales características y deficiencias de los mismos. Por último se realiza un análisis de las metodologías de desarrollo de software, herramientas y lenguajes a utilizar para modelar la solución.

Capítulo 2: Negocio y Análisis del sistema.

Se realiza una descripción más detallada de la situación problemática existente en el entorno donde se realizará la captura de requisitos. Se analizan y modelan los procesos de negocio. Se identifican las técnicas a utilizar para la captura de requisitos, los requerimientos asociados a los procesos y se describen los mismos. Se definen las técnicas utilizadas para la validación de requisitos así como las métricas de calidad de la especificación de éstos.

Capítulo 3: Diseño del sistema.

Se realiza el diseño del sistema. Se identifica y confecciona el diagrama de clases del diseño, se realiza el diseño de la base de datos y se obtienen otros artefactos correspondientes al Diseño definidos en la metodología de desarrollo seleccionada. Se identifican los patrones de diseño empleados en la solución.

Capítulo 4: Validación del sistema.

Se realiza la validación de los artefactos generados a través de técnicas de validación de requisitos y la aplicación de métricas orientadas a la calidad de las especificaciones de los requisitos y al modelo de diseño.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se brindan los fundamentos teóricos de la investigación, se describen los principales conceptos empleados en el documento para facilitar una mejor comprensión del tema. Se presentan los resultados de un estudio realizado a Sistemas de Trámites Migratorios. Por último, se analizan diferentes lenguajes, tecnologías y metodologías de desarrollo de software y otros aspectos importantes a emplear en el desarrollo del trabajo.

1.2 Marco conceptual

La cooperación internacional en la universidad se desarrolla a través de la participación de estudiantes, profesores y trabajadores en organismos y organizaciones internacionales, en programas y proyectos de investigación internacional, en redes académicas y/o tecnológicas, en eventos científicos, concursos, estudios de superación y en intercambios oficiales.

La **Dirección de Cooperación Internacional (DCI)** es la estructura organizativa que se encarga de la gestión de las actividades de colaboración internacional y relaciones públicas en la UCI. Pertenece a la Vicerrectoría de Investigaciones y Cooperación Internacional (1); la misma está compuesta por el Grupo de Cooperación Internacional, Grupo de Relaciones Públicas y Grupo de Trámites. El primero es el encargado de gestionar la estrategia de colaboración internacional de la universidad, coordinar la participación de la UCI en organizaciones internacionales de excelencia, diseñar y fomentar los vínculos inter- institucionales y contribuir a la identificación y gestión de programas de cooperación internacional.

(1) Por otra parte el Grupo de Relaciones Públicas se encarga de manejar todo lo referente a la imagen institucional de la universidad así como de la promoción y divulgación de actividades de interés del centro. Conjuntamente con el Grupo de Cooperación Internacional, dirige metodológica y funcionalmente el sistema de colaboración internacional de la UCI, gestiona y conforma la información de apoyo a los intercambios inter- institucionales además de garantizar las acciones necesarias para el éxito de las misiones de la Universidad en el exterior y la satisfacción de los visitantes en Cuba. (1)

Grupo de Trámites

Pertenece a la DCI, es el encargado de dirigir los asuntos migratorios y de extranjería en la universidad. Gestiona la documentación necesaria para que una persona pueda viajar al exterior o un

extranjero pueda entrar al país por motivos de trabajo relacionados con la UCI. Elabora los reportes estadísticos y económicos de las personas que viajan y de las misiones, además de los partes sobre el estado de estas últimas y del proceso de tramitación de los misioneros. Controla y archiva los documentos y la información obtenida de dichos procesos. (1)

Misión de Cooperación Internacional: Es una inversión a mediano o largo plazo para alcanzar objetivos estratégicos de la organización a través de la cooperación internacional. (2)

En la universidad se encuentran definidas de la siguiente forma:

Tipo I- Oficial: consiste en el viaje al exterior de algún directivo de la universidad para intercambiar con instituciones extranjeras. (2)

Tipo II- Comercial: Son las misiones cuyo objetivo es comercializar, explorar, promover, comprar o vender productos y/o soluciones informáticas o firmar contratos con este fin. (2)

Tipo III- Evento: Consisten en seminarios, conferencias, congresos que se realizan en el exterior. También se consideran eventos los concursos en los que fundamentalmente participan los estudiantes. (2)

Tipo IV- Proyecto: Misiones que generalmente son financiadas por el proyecto de desarrollo las cuales se realizan en función de los objetivos del mismo. Estos proyectos se clasifican en tres tipos: I+D², Productivos (Misión que financia o financiará el proyecto) y Colaboración (con otros países, organismos, entidades. Ejemplo: CDI³, Discapacitados, Correos de Cuba, entre otros.) (2)

Tipo V- Estancias: Viaje al exterior de profesores por un período determinado que puede extenderse desde una semana a 6 meses por motivos de: Curso, Diplomado, Maestría, Doctorado, Post doctorado, Intercambio académico, Investigación, Profesor invitado y Entrenamiento. (2)

² Investigación más Desarrollo

³ Centros de Diagnóstico Integral

1.3 Sistemas de Trámites Migratorios

Los sistemas de trámites migratorios contribuyen a la mejora de los procesos de gestión de documentación que requiere un trámite, optimizando la administración de los recursos materiales y humanos. (3)

Debido a la necesidad de informatizar el proceso de Gestión de Trámites de Salida al Exterior que se realiza en la Facultad 3 se estudiaron Sistemas de Trámites Migratorios existentes a nivel nacional a fin de poder determinar si se podía ajustar alguno de ellos a las necesidades actuales en este nivel. Durante la investigación no se tuvieron en cuenta los sistemas internacionales existentes ya que cada país establece sus propias leyes relativas a los trámites para la salida al exterior en correspondencia con la política de relaciones internacionales adoptada por lo que se llegó a la conclusión de que ningún sistema informático que de procedencia extranjera puede servir como base o guía para el desarrollo del presente trabajo.

1.3.1 Sistemas de Trámites Migratorios en Cuba

El **Ministerio de la Industria Sideromecánica (SIME)** cuenta con un Sistema de Atención a Misiones que se utiliza desde el año 2000, fue elaborado por la Empresa de Servicios Informáticos de Pinar del Río. Se desarrolló en el plazo de 4 meses sin la utilización de alguna metodología de desarrollo, basándose solo en la interacción programador- cliente y las entrevistas realizadas para conocer los requisitos. El sistema es una aplicación de escritorio realizada en Delphi, presenta grandes beneficios a esta empresa y le brinda a la misma algunos servicios como la búsqueda de información de los misioneros que han viajado y de las misiones realizadas, búsqueda sobre el estado de los pasaportes, generación de los modelos oficiales de Solicitud de Trámites, Pasaporte y Visa de la Dirección de Inmigración y Extranjería además de brindar un sistema de reportes. (3)

El uso de este sistema ahorra tiempo y fuerza de trabajo a los responsables de tramitar misioneros del SIME pero por otro lado, desde hace varios años la Empresa de Servicios Informáticos de Pinar del Río no brinda soporte a este sistema por lo que en la actualidad las necesidades de la empresa no son cubiertas totalmente por la aplicación. Una de las necesidades de este ministerio es que las empresas y grupos empresariales adjuntos al mismo utilicen una aplicación con el objetivo de tener toda la información en la misma base de datos, la misma pudiera ser satisfecha si el sistema fuera una aplicación web. En la

actualidad esta empresa realiza una serie de trámites a extranjeros que el sistema no cubre por completo ya en el momento en que fue desarrollado estas gestiones no se realizaban. (3)

Por otra parte el **Ministerio de Educación Superior (MES)** cuenta con un sistema de gestión para la cooperación internacional que opera sobre web e incluye una potente herramienta para realizar cualquier tipo de reporte. Presenta el inconveniente de que por cada cliente que hace una petición al sistema se ejecuta una aplicación diferente en el servidor, lo cual no constituye un problema a una empresa o institución pequeña y con recursos limitados pero es un inconveniente para la UCI donde pueden haber un gran número de usuarios conectados simultáneamente al sistema. Por otra parte la base de datos está diseñada en Microsoft Access, gestor que no ofrece mucha seguridad en la protección de la información. (3)

El **Ministerio de la Informática y las Comunicaciones (MIC)** cuenta con un sistema web que incluye la gestión y aprobación de solicitudes a nivel de Ministerio y procesos homólogos a los desarrollados por el Grupo de Cooperación Internacional de la DCI en la UCI. (3)

La **Universidad de las Ciencias Informáticas** cuenta con un Sistema Informatizado de Relaciones Internacionales el cual surge con el objetivo de informatizar los procesos de la DCI de la UCI. En el año 2006, la aplicación web desarrollada con este propósito, tenía disponibles las funcionalidades dedicadas a soportar los procesos del Grupo de Trámites. Su implementación se realizó utilizando como plataforma de desarrollo Visual Studio .NET 2003 y Microsoft SQLServer 2000 como gestor de base de datos. Hasta esa fecha se había llegado hasta la fase de análisis de los procesos de los dos restantes grupos de la DCI. Como resultado de un cambio en las políticas de la UCI en cuanto a la selección de herramientas para la informatización del centro hacia la migración a tecnologías libres, fue necesario rediseñar la aplicación existente, adaptándola a la nueva arquitectura y se comenzó a redefinir el trabajo realizado como paradigma la soberanía tecnológica. (3)

1.3.2 Valoraciones de los sistemas estudiados

El estudio de algunos sistemas existentes en el país que permiten la gestión de trámites migratorios pero no se ajusta a las necesidades de la facultad. Lo anteriormente planteado se basa en lo siguiente: el uso del sistema existente en el **SIME** no es conveniente en la Universidad pues es una aplicación de escritorio lo cual constituye una gran desventaja pues sería necesario instalar un cliente en cada PC que necesite

conectarse a este sistema, lo cual no se ajusta a la política de informatización de la UCI. Por otra parte la estructura de Dirección de Cooperación Internacional del **MES** no coincide con la organización de la existente en la DCI de la UCI así como tampoco los tipos de misiones y la clasificación de las mismas por lo que los procesos que se realizan para tramitar la salida al exterior no coinciden con los que se desarrollan en la Universidad. El sistema web existente en el **MIC** comparte procesos similares a los que desarrolla el Grupo de Cooperación Internacional perteneciente a la DCI en la universidad pero no posee ningún tipo de documentación con la cual se pueda hacer un estudio profundo del mismo con el propósito de adaptarlo a la Facultad 3, además la realización de este trabajo se enfoca más a las actividades que realiza el Grupo de Trámites. Por último se realizó un análisis de un sistema existente en la **Universidad de las Ciencias Informáticas**, mediante el cual se realizan actividades relacionadas con la gestión de trámites de salida al exterior a través de procedimientos establecidos por la dirección del centro para dicha gestión. La Facultad 3, al pertenecer a la Universidad se acoge a lo establecido por la Dirección de Cooperación Internacional para gestionar los trámites necesarios para la salida al exterior una vez que es solicitada una misión, pero no puede hacer uso del sistema en cuestión ya que éste no contempla entre sus funciones las actividades que se llevan a cabo previamente en cada una de las facultades, solamente se encarga de la gestión de un grupo de informaciones y documentación una vez que la solicitud de la misión es aprobada a nivel de facultad.

1.4 Aplicaciones Web

Una aplicación Web será un sistema Web donde la entrada del usuario (entrada de datos y navegación) afecta el estado del negocio. (4)

Las aplicaciones web trabajan sobre la base de la arquitectura Cliente Servidor, la cual es una nueva tendencia en el desarrollo de redes, que tiene como objetivo optimizar el uso tanto del hardware como del software a través de la separación de funciones: el cliente, quien inicia una determinada petición y el servidor, dedicado a responder dichas peticiones.

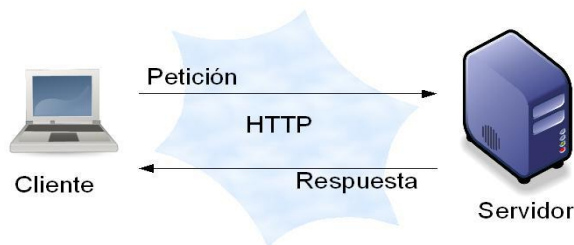


Figura 1 Arquitectura Cliente-Servidor

Las aplicaciones web presentan una ventaja significativa respecto al desarrollo de aplicaciones de escritorio. La misma consiste en su forma de instalación y distribución (4) lo cual ha hecho optar por el desarrollo de una aplicación de este tipo para gestionar los trámites necesarios para la salida al exterior que tienen lugar en la Facultad 3. Se fundamenta esta elección ya que puede ser accedida vía web por una internet o una intranet y el acceso a ésta puede ser público o restringido dependiendo hacia quién vaya dirigida. Las actualizaciones y el mantenimiento de este tipo de aplicaciones son transparentes para los usuarios ya que éstos no necesitan instalar ni actualizar software en sus computadoras. Se caracterizan por ser multiplataforma ya que pueden ejecutarse en cualquier Sistema Operativo que posea un navegador web, además permite que existan varios clientes conectados a una única aplicación instalada en un servidor por lo que se puede actualizar y mantener una sola aplicación a través de la cual sus clientes verán los resultados inmediatamente.

1.5 Metodología de desarrollo

La metodología de desarrollo, dentro de la ingeniería de software, se encarga de elaborar estrategias de desarrollo de software a fin de promover prácticas centradas en las personas o equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieran implicación directa con el cliente. Su objetivo es convertir el desarrollo de software en un proceso formal, con resultados predecibles que permitan obtener un producto de alta calidad, que satisfaga las necesidades y expectativas del cliente. (5)

En un proyecto de desarrollo de software, la metodología define “quién” debe hacer “qué”, “cuándo” y “cómo” debe hacerlo para alcanzar un determinado objetivo. (6)

Existen las Metodologías Tradicionales y las Metodologías Ágiles. Las primeras intentan conseguir un objetivo común por medio de orden y documentación mientras que las llamadas Ágiles tratan de mejorar la

calidad del software por medio de una comunicación directa e inmediata entre las personas que intervienen en el proceso de desarrollo.

1.5.1 Principales metodologías de desarrollo

Dentro de las metodologías más utilizadas mundialmente figuran por parte de las **Metodologías Ágiles:** Extreme Programming (XP) y SCRUM. Entre las **Metodologías Tradicionales:** Rational Unified Process (RUP) y Microsoft Solution Framework (MSF). A continuación se describen algunas características de estas metodologías identificadas durante el estudio realizado a fin de seleccionar la más adecuada que se ajuste a las necesidades existentes.

Rational Unified Process

RUP fue creada por el mismo grupo de expertos que crearon UML, Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1998. El objetivo que se perseguía con esta metodología era producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos (7), esta metodología concibió desde sus inicios el uso de UML como lenguaje de modelado. Es un proceso dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo (requisitos, análisis, diseño, implementación, prueba) que parten de los casos de uso; está centrado en la arquitectura y es iterativo e incremental. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software.

Microsoft Solution Framework

MSF se caracteriza por ser una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. Además es un proceso adaptable, escalable y flexible. MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y el Modelo de Aplicación. (8)

Extreme Programming (XP)

Desarrollada por Kent Beck, está centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de

los desarrolladores, y propiciando un buen clima de trabajo. XP está guiada por una rápida programación y se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, en la reutilización de código, en la realización de pruebas a los principales procesos con el objetivo de tratar de obtener los posibles errores futuros, conocido como pruebas unitarias, en la simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Impone un alto nivel de disciplina entre los programadores, lo cual permite mantener un mínimo nivel de documentación que a su vez se traduce en una gran velocidad de desarrollo, además de proponer que el trabajo de los programadores sea en pares de forma tal que uno realice lo que el otro no hace en ese instante. (7)

SCRUM

SCRUM define un proceso empírico, iterativo e incremental de desarrollo que intenta obtener ventajas respecto a los procesos definidos (cascada, espiral, prototipos, etc.) mediante la aceptación de la naturaleza caótica del desarrollo de software, y la utilización de prácticas tendientes a manejar la impredecibilidad y el riesgo a niveles aceptables.

Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos, el desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. (8)

1.5.2 Valoración de las metodologías estudiadas

RUP provee un entorno de proceso de desarrollo configurable basado en estándares que permite tener claro y accesible el proceso de desarrollo que se sigue y que éste sea configurado a las necesidades de la organización y del proyecto. A pesar de proveer de documentación detallada de todo cuanto se realiza durante la construcción del software no puede ser aplicada como guía durante la construcción de la aplicación que se propone en el presente trabajo ya que el desarrollo de la misma se realiza a partir de procesos de negocio, lo cual no se ajusta a los principios de RUP que es dirigido por casos de uso. Por otra parte MSF basa su desarrollo con tecnología Microsoft lo que limita las opciones del cliente en lo que

se refiere a herramientas de desarrollo. Esta metodología hace un análisis de riesgo demasiado exhaustivo que puede frenar el avance del proyecto y además solicita demasiada documentación en todas las fases resultando muy engorroso el trabajo y por consiguiente afecta el tiempo de entrega del producto. Por otra parte, tanto SCRUM como XP son metodologías que son especialmente adecuadas para proyectos con requisitos imprecisos y muy cambiantes (8) lo cual descarta su uso ya que los requisitos asociados al presente trabajo se encuentran bien definidos, el cliente no forma parte del equipo de desarrollo y se requiere la existencia de documentación de todo el proceso de desarrollo de la aplicación que permita a otros equipos de desarrollo comprender las actividades que se llevaron a cabo durante su construcción.

1.6 Modelo de desarrollo adoptado

Para guiar el proceso de desarrollo de la aplicación propuesta se decidió adoptar por el modelo de desarrollo personalizado y definido por la dirección del proyecto ERP- Cuba el cual se nutre de los aspectos positivos y ágiles de varias de las metodologías conocidas dentro del mundo del desarrollo del software (RUP, XP, Orientada a Componentes). Este modelo nace a partir de la necesidad de dicho proyecto de que cada uno de los equipos de desarrollo posean un modelo estandarizado, así como una definición clara y precisa de las responsabilidades de cada uno de los roles involucrados en el desarrollo de la solución. (9)

La elección de este modelo de desarrollo se fundamenta en que permite modelar el negocio a través de procesos, la ingeniería de requisitos es mucho más clara que en otras metodologías. Es orientado a componentes lo cual posibilita la independencia de funciones del sistema a la hora de mantener o modificar el sistema funcional. Además admite como lenguaje de modelado del negocio BPMN y para el diseño UML respectivamente. El uso de este modelo de desarrollo permitirá la generación de artefactos de vital importancia durante el análisis y el diseño como son: Modelo de proceso de negocio, Descripción de procesos de negocio, Modelo conceptual, Prototipos de interfaz de usuario, Especificación de requisitos, Diagrama de clases, entre otros.

Para conocer más detalles acerca de los roles y actividades del Modelo de Desarrollo así como también del flujo de éstas se pueden consultar los Anexos 1, 2 y 3 del presente trabajo.

1.7 Lenguajes de Modelado

Notación para el Modelado de Procesos del Negocio

La Notación para el Modelado de Procesos del Negocio (BPMN: Business Process Modeling Notation) es un estándar de modelado de procesos de negocio (10) donde se presentan gráficamente las diferentes etapas del proceso del mismo. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes. Es muy utilizado a nivel mundial y considerado un estándar avanzado. BPMN ayuda a definir, documentar y hacer mucho más eficientes los procesos del negocio. Su principal objetivo es proveer una notación estándar facilitando el entendimiento de las colaboraciones y transacciones de negocio para todos los involucrados e interesados del negocio. Entre estos interesados están los analistas de negocio (quienes definen y redefinen los procesos), los desarrolladores técnicos (responsables de implementar los procesos) y los gerentes y administradores del negocio (quienes monitorean y gestionan los procesos). Tiene como finalidad servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación. (10) En BPMN el modelado es realizado mediante diagramas muy simples con un conjunto muy pequeño de elementos gráficos. Con esto se busca que para los usuarios del negocio y los desarrolladores técnicos sea fácil entender el flujo y el proceso.

Las cuatro categorías básicas de elementos son:

Objetos de flujo: Eventos, Actividades, Rombos de control de flujo (Gateways)

Objetos de conexión: Flujo de Secuencia, Flujo de Mensaje, Asociación

Swimlanes (Carriles de piscina): Pool, Lane

Artefactos: Objetos de Datos, Grupo, Anotación

Con estas cuatro categorías de elemento se puede realizar un diagrama simple de procesos de negocio (BPD: Business Process Diagram).

Lenguaje de Modelado Unificado

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimientos sobre los sistemas que se deben construir. Se usa para entender, diseñar, configurar,

mantener, y controlar la información sobre tales sistemas. Está pensado para emplearse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. (11)

1.7.1 Comparación entre UML y BPMN

El Lenguaje de Modelado Unificado (UML) toma un perfil orientado a objetos en el modelado de aplicaciones, mientras que la Notación para el Modelado de Procesos del Negocio (BPMN) toma un perfil orientado a procesos en el modelado de sistemas. Además BPMN tiene un enfoque específico a los procesos de negocio y UML se enfoca más al diseño del software; siendo ambas notaciones totalmente compatibles entre sí. Las extensiones de UML para el modelado de negocio aportan elementos muy importantes ya que proporcionan algunas otras vistas de la arquitectura de negocio que son más difíciles de observar usando únicamente BPMN. Por ejemplo, la visualización de las responsabilidades de los trabajadores del negocio, la manipulación de las entidades del negocio y la comprensión de los estados asociados a las entidades del negocio. Por las razones antes expuestas el modelado del negocio se realizará utilizando BPMN y para el diseño el Lenguaje de Modelado Unificado, aunque como también se ha visto UML puede ser utilizado en el modelado del negocio.

1.8 Herramientas de Modelado

El múltiple desarrollo de software ha influido grandemente en la aparición de novedosas herramientas automatizadas para ayudar a los diseñadores a realizar el proceso de análisis y diseño del software, haciendo posible que tenga una mejora en la calidad, fiabilidad, utilidad y rendimiento del mismo. Una de las herramientas más utilizadas en estos tiempos son las herramientas CASE (Computer Aided Software Engineering) o Ingeniería de Software Asistida por Computadora.

Una Herramienta CASE es un conjunto de aplicaciones informáticas que tienen el objetivo de aumentar la productividad en el desarrollo de un software mitigando los costes en términos de tiempo. (12) La utilidad de estas herramientas consiste principalmente en realizar un buen diseño del proyecto y a partir de este, implementar parte del código automáticamente, documentar o detectar errores, entre otras que hacen de ellas una fuente importante para la creación de un producto con calidad.

1.8.1 Visual Paradigm for UML 6.1 Enterprise Edition

Visual Paradigm para UML, es una potente herramienta visual UML CASE. Visual Paradigm UML está diseñado para una amplia gama de usuarios. Esta herramienta además de soportar modelado UML, provee el modelado de procesos de negocio Business Process Modeling Notation, en su edición Enterprise Edition soporta la última versión de BPMN, la cual es una sencilla pero potente notación para el modelado de flujos de trabajo de un negocio. Otras de las atractivas características de Visual Paradigm es que permite la sincronización entre Diagramas Entidad Relación y Diagramas de Clases, la generación de Código/Ingeniería Inversa. (13) Es además una herramienta multiplataforma.

1.9 Sistema Gestor de Contenido

“Los sistemas de gestión de contenido (Content Management System o CMS) es un software que se utiliza principalmente para facilitar la gestión web, ya sea en Internet o en una intranet, y por eso también son conocidos como gestores de contenido web (Web Content Management o WCM).” (14) Su principal objetivo es proveer al desarrollador de una herramienta para la construcción de aplicaciones web que manejen contenidos de forma dinámica y que para el desarrollo de la misma no requieran de amplios conocimientos técnicos referentes a la programación.

Actualmente existen muchos CMS, entre ellos Mambo, Plone, Muravey-Web, Typo3, Drupal y Joomla. La aplicación que se propone desarrollar requiere de tecnología multiplataforma y de código abierto para garantizar usos posteriores de la misma por lo que se ha considerado solamente el análisis que aquellos que sean de código abierto, programados en el lenguaje seleccionado PHP y que brinden soporte para el desarrollo de portales. Por tal motivo se realizará el análisis de Drupal y Joomla respectivamente a fin de determinar el más idóneo a utilizar. A continuación se describen algunas características de los mismos.

Drupal: es una plataforma dinámica para la construcción de sitios Web que permite a un individuo o una comunidad de usuarios publicar, manejar y organizar una variedad de contenido. Drupal integra muchas

características populares de los Sistemas de Gestión de Contenido, weblogs, herramientas de colaboración y comunidad de discusión, todo en un solo paquete fácil de utilizar. Como software de código abierto desarrollado y mantenido por una comunidad Drupal es libre para descargarlo de Internet y usarlo. (15)

Joomla: corresponde al grupo de soluciones de código abierto, es un producto de software libre. Permite organizar eficientemente los contenidos de un sitio en secciones y categorías, lo que facilita la navegabilidad para los usuarios y permite crear una estructura sólida, ordenada y sencilla para los administradores. Construido también bajo licencia GPL. Fue creado en PHP y usa base de datos MySQL. Joomla incluye características como: hacer caché de páginas para mejorar el rendimiento Web, flash con noticias, foros y calendarios. (16)

Seguidamente, a pesar de las amplias ventajas que poseen estos dos CMS en cuanto a su portabilidad, gestión de usuarios, de contenidos, entre otros; se fundamenta el uso del Joomla como sistema gestor de contenido elegido para el desarrollo de esta solución.

El CMS Joomla presenta entre sus principales virtudes la de permitir editar el contenido de un sitio web de manera sencilla. Es una aplicación de código abierto, programada mayoritariamente en PHP⁴ bajo una licencia GPL⁵. Consume pocos recursos de la PC servidor y los inconvenientes que pueden existir referentes a la licencia de la base de datos que utiliza son solubles ya que ORACLE, la compañía propietaria de esta licencia, solo las requiere cuando se va a comercializar el producto o si su uso es con fines privativos pero este no es el caso de la aplicación que se propone. Se fundamenta el uso de esta herramienta basado en que la aplicación a desarrollar puede ser desarrollada íntegramente utilizando este CMS, se cuenta con el apoyo de la Comunidad de Joomla de la UCI que contribuye al soporte de la aplicación a desarrollar y además por todas las características anteriormente expuestas sobre este sistema gestor de contenido.

Se hará haciendo uso de la versión estable de Joomla 1.6, publicada el 10 de enero de 2011 que incluye numerosas mejoras entre las que cabe contar: mejor control de acceso para el visionado y edición del

⁴ Acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

⁵ Acrónimo en inglés de General Public License (Licencia Publica General). Regula los derechos de autor de los programas de software libre.

contenido con grupos de usuario y niveles de acceso configurables, presenta una estructura de categorías definibles por el usuario desde un simple nivel hasta múltiples niveles complejos de categorías. Mejoras de instalación que permiten la instalación de múltiples extensiones desde dentro de un paquete y que se actualiza con solo un clic. Brinda soporte de idiomas expandido para una fácil producción de sitios multi-idioma. Ofrece además un mayor control creativo mediante los estilos de las nuevas plantillas y mejoras en el flujo de trabajo y productividad entre otras características adicionales. (16)

1.10 Sistema Gestor de Base de Datos

Sistema Gestor o Manejador de Bases de Datos (SGBD) es un conjunto de programas que permite a los usuarios crear y mantener una BD, por lo tanto, el SGBD es un software de propósito general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones. (17)

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. (18)

El CMS Joomla puede trabajar en Internet o Intranet y requiere de una base de datos MySQL por lo que el SGBD a utilizar para el desarrollo de la aplicación propuesta es MYSQL 5.0.5.

1.11 Servidor Web

Un servidor web básicamente, sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP (Hypertext Transfer Protocol).

Un servidor web Apache es una tecnología gratuita de código fuente abierta, puede ser usado en varios sistemas operativos, lo que lo hace prácticamente universal. Es un servidor altamente configurable, es decir, se pueden elegir qué características van a ser incluidas en el servidor seleccionando que módulos se van a cargar, ya sea al compilar o al ejecutar el servidor. (19)

Debido a que el servidor web Apache, es flexible, robusto y teniendo en cuenta que Joomla requiere preferiblemente un servidor de este tipo, se utilizará como servidor web en el posterior desarrollo de la aplicación.

1.12 Lenguajes de programación

Desde el inicio de la red de redes Internet, surgieron diferentes demandas por los usuarios dándose soluciones a éstas mediante lenguajes estáticos. Con el transcurso del tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución, dando lugar a desarrollar lenguajes de programación para la web dinámica, que permitieran interactuar con los usuarios y utilizaran sistemas de bases de datos. A continuación se describen los lenguajes de programación del lado del cliente y del servidor que se emplearán en el desarrollo de la solución.

1.12.1 Lenguajes de programación del cliente

Los lenguajes del lado del cliente son aquellos que pueden ser directamente interpretados por el navegador.

HTML

El HTML (Hypertext Markup Language) es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto que es el formato estándar de las páginas Web.

Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML. (20)

Java Script

Java Script es el lenguaje de scripts más utilizado hoy en la Web del lado del cliente. Es ampliamente usado en tareas que van desde la validación de formularios de datos a la creación de complejas interfaces de usuario. También puede ser utilizado para manipular las marcas en el documento en el cual está contenido.

CSS

Cascading Style Sheets (Hojas de Estilo en Cascada). Es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación. CSS es una especificación desarrollada por el W3C (World Wide Web Consortium) para permitir la separación de los

contenidos de los documentos escritos en HTML, XML, XHTML, SVG, o XUL de la presentación del documento con las hojas de estilo, incluyendo elementos tales como los colores, fondos, márgenes, bordes, tipos de letra..., modificando la apariencia de una página web de una forma más sencilla, permitiendo a los desarrolladores controlar el estilo y formato de sus documentos.

1.12.2 Lenguajes de programación del servidor

Los lenguajes del lado del servidor que son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Responden a las peticiones del usuario, y generan páginas Web dinámicas. A continuación se describe el lenguaje del lado del servidor a emplear en el desarrollo de la aplicación web.

PHP 5

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. Es usado principalmente en interpretación del lado del servidor. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. (21) Es un lenguaje libre por lo que se presenta como una alternativa de fácil acceso para todos y además multiplataforma siendo esto una de las principales características por lo que es muy usado en el mundo.

1.13 Entorno Integrado de Desarrollo

Un entorno de desarrollo integrado o IDE (en inglés: Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación que permite de forma cómoda y ágil editar, compilar, ejecutar y depurar programas.

El IDE empleado para el desarrollo de la aplicación es el NetBeans 6.9, un entorno integrado de desarrollo galardonado disponible para Windows, Mac, Linux y Solaris, de código libre y una plataforma de aplicaciones que permite a los desarrolladores crear rápidamente web, empresa, escritorio y aplicaciones móviles utilizando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, Ruby y Ruby on Rails, Groovy y Grails, y C / C ++.

1.14 Marcos de Trabajo

Los marcos de trabajo o framework, como también se les conoce, no son más que arquitecturas definidas para un determinado dominio de la aplicación que contiene un conjunto de componentes implementados y sus interfaces bien definidas, estos componentes se pueden utilizar, redefinir y crear nuevos componentes. Los objetivos principales que persigue un framework son, entre otros, acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

ExtJs framework: es un framework para Java Script muy utilizado en el desarrollo de aplicaciones Web con AJAX. Tiene una librería inmensa que permite configurar las interfaces Web de manera semejante a aplicaciones de escritorio. Contiene la mayoría de los controles de los formularios Web incluyendo Grids para mostrar datos y elementos semejantes a la programación de escritorio como los formularios, paneles, barras de herramientas, menús y muchos otros. Dentro de su librería de componentes incluye elementos para el manejo de datos, lectura de XML, lectura de datos JSON e implementaciones basadas en AJAX.

1.15 Conclusiones

Tras haber concluido el estudio de los sistemas antes expuestos se puede resumir que no existe un sistema que sirva como guía para el desarrollo de la solución a la problemática planteada y que presente entre sus funciones algunas similares a las que se desarrollan en la facultad como parte de la gestión de trámites de salida al exterior de un individuo. El único que, por ser desarrollado en la universidad, tiene más semejanza con los procesos de gestión de cooperación internacional es el utilizado por el Grupo de Trámites de la Dirección de Cooperación Internacional en la UCI pero no cuenta entre sus funcionalidades las necesarias para realizar la solicitud de las misiones y la aprobación de las mismas en la Facultad 3. La metodología de desarrollo, lenguajes y herramientas que fueron objeto de estudio y posteriormente seleccionadas se adaptan a las necesidades actuales del proyecto. Por último, el análisis de las características expuestas en el capítulo permite afirmar que el correcto uso de las mismas garantizará la construcción de un sistema que resuelva la problemática planteada.

CAPÍTULO 2: NEGOCIO Y ANÁLISIS DEL SISTEMA

2.1 Introducción

En el presente capítulo se realiza el modelamiento del negocio a fin de comprender la estructura y la dinámica de la organización donde se va a implantar el sistema a través de la descripción de los procesos de negocio identificados asegurando con ello un mejor entendimiento entre el cliente y el equipo de desarrollo. Una vez concluido este paso se identifican y describen los requisitos de software que fueron capturados y se procede con el desarrollo de un grupo de artefactos correspondientes al análisis del sistema los cuales serán descritos durante el desarrollo del capítulo.

2.2 Modelamiento del Negocio

El primer paso en el proceso de desarrollo de software es precisamente alcanzar cierto nivel de conocimientos sobre el problema en cuestión. Un sistema, por pequeño que sea, generalmente es complicado. Por eso se necesita dividirlo en piezas si se pretende comprenderlo y gestionar su complejidad. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales. (22)

Una técnica para la especificación de los requisitos más importantes del sistema, que da soporte al negocio, es el modelo del negocio, con lo cual se refuerza la idea de que sea el propio negocio lo que determine los requisitos. (22) De ahí, que en el campo del software también resulte útil la creación de modelos que organicen y presenten los detalles importantes de problemas reales que se vinculan con el sistema informático a construir. Estos modelos deben cumplir una serie de propiedades, entre ellas la de ser coherentes y relacionados. Uno de los modelos útiles previo al desarrollo de un software es el modelo del negocio.

El modelamiento del negocio se realiza con el objetivo comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema a fin de asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.

2.3 Procesos de Negocio

Se puede definir un proceso como una serie de actividades, en las que varias entidades (personas, máquinas, etc.) colaboran para conseguir un objetivo concreto. Un proceso del negocio es el conjunto

estructurado de las actividades que han sido diseñadas para producir un resultado específico para un cliente o el mercado. (22) El principal objetivo de los procesos de negocio es satisfacer necesidades de los clientes.

Un proceso de negocio es un conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido. Describen como es realizado el trabajo en la organización y se caracterizan por ser observables, medibles, mejorables y repetitivos.

En la Facultad 3 se realizan varios procesos, entre ellos se encuentra la Gestión de Trámites de Salida al Exterior. Este proceso tiene como objetivo gestionar los trámites referentes a las solicitudes de misiones de cooperación internacional que se realizan desde las diferentes áreas de la facultad. Para comprender cada una de las actividades que se deben desarrollar para tramitar una solicitud de misión se realizaron varias entrevistas con el Asesor de Cooperación Internacional. Como resultado de las mismas se obtuvo el conocimiento de que las misiones de cooperación internacional que se encuentran definidas en la Universidad pueden ser solicitadas por dos vías; que el individuo o la persona interesada realice la solicitud directamente o que sea solicitada por la organización. La primera requiere necesariamente su presentación y consulta ante el área de administración a la cual pertenece el solicitante, la evaluación por parte de los órganos de consulta (Consejo de Investigación- Posgrado-Cooperación) y del Asesor de Cooperación Internacional para poder continuar con el proceso de tramitación de la solicitud. La segunda en cambio, no precisa de su análisis por parte de los canales anteriormente descritos e iniciaría su tramitación en la facultad a partir de la creación de la planilla de solicitud que posteriormente pasa a ser aprobada en el Consejo de Dirección.

Por todo lo antes explicado, el proceso de Gestión de Trámites de Salida al Exterior que tiene lugar en la facultad se compone de los 3 subprocesos siguientes, los cuales aparecen ilustrados en la Figura 2.

- Presentar propuesta de solicitud de misión.
- Crear planilla de solicitud de misión
- Aprobar solicitud de misión.

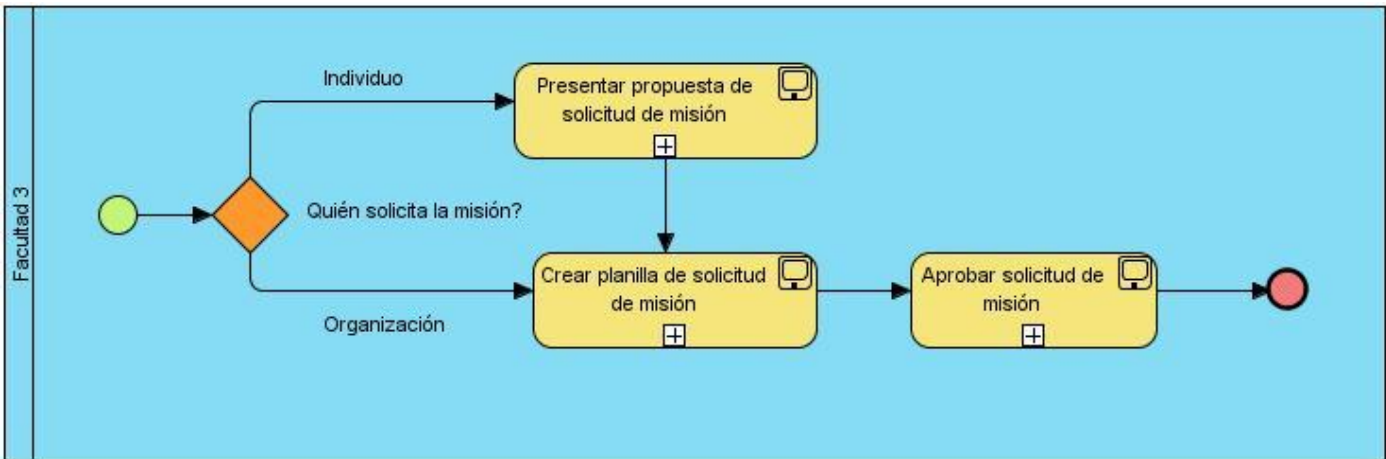


Figura 2 Gestionar Trámites de Salida al Exterior

Nombre:	Gestionar Trámites de Salida al Exterior.
Misión:	Gestionar los trámites referentes a la solicitud de misión de cooperación internacional para la salida al exterior de un individuo.
Alcance:	Presentar, validar, evaluar y aprobar la solicitud de misión de cooperación internacional.
Subprocesos:	Presentar propuesta de solicitud de misión. Crear planilla de solicitud de misión. Aprobar solicitud de misión.

2.4 Descripción de Procesos de Negocio

A continuación se describen brevemente cada uno de los procesos identificados y se muestran los diagramas asociados a éstos donde se establece la secuencia de pasos a seguir para su realización.

2.4.1 Descripción del proceso Presentar propuesta de solicitud de misión.

La persona interesada en solicitar una misión de cooperación internacional, en este caso el solicitante, pone en conocimiento su interés ante el Área de Administración a la cual pertenece haciéndole llegar ciertos datos de consulta (datos personales del solicitante, tipo de misión que solicita y el motivo por el cual realiza la solicitud) para que sean validados. Dichos datos son consultados por diferentes canales

que determinan si la solicitud presentada constituye una posible misión y dependiendo del resultado permite que se continúe con el proceso de tramitación de la propuesta presentada.

Para obtener una descripción más detallada del flujo básico de actividades consultar el Anexo 4.

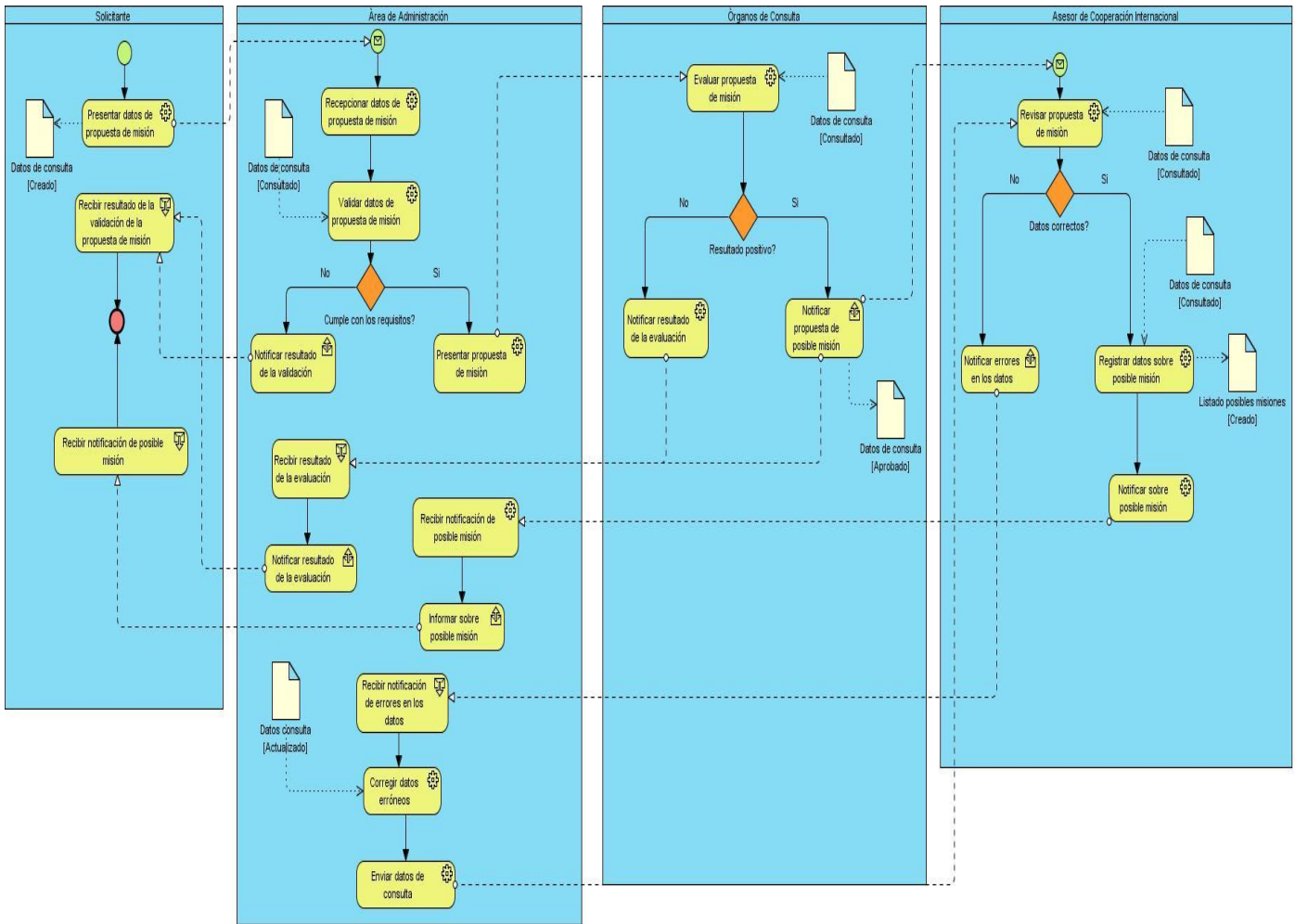


Figura 3 Diagrama del proceso Presentar propuesta de solicitud de misión

2.4.2 Descripción del proceso Crear planilla de solicitud de misión

Una vez que la propuesta de misión presentada por el solicitante es validada como posible misión de cooperación internacional se procede a la creación de la planilla de solicitud de misión que más adelante es presentada y aprobada en el Consejo de Dirección de la Facultad, constituyendo éste uno de los procesos posteriores a la presentación de la propuesta de misión más importante durante la tramitación de la salida al exterior por concepto de misión. Durante este proceso se le pide al solicitante una serie de documentos complementarios a la solicitud (fotos, imágenes de ambas caras del CI y una serie de avales y certificaciones que son relativos al tipo de misión que se está solicitando, entre otros) con los cuales, apoyándose en los datos de consulta anteriormente presentados, el Área de Administración a la cual pertenece el solicitante procede a confeccionar la Planilla de solicitud de misión. Dicha planilla es enviada al Asesor de Cooperación Internacional para su revisión. Una vez revisada satisfactoriamente dicha planilla queda pendiente a su aprobación por parte de la dirección de la facultad. A través de la realización de este proceso quedan plasmados claramente los datos necesarios para que la solicitud pueda ser presentada y analizada en el Consejo de Dirección de la facultad.

Para obtener una descripción más detallada del flujo básico de actividades consultar el Anexo 5.

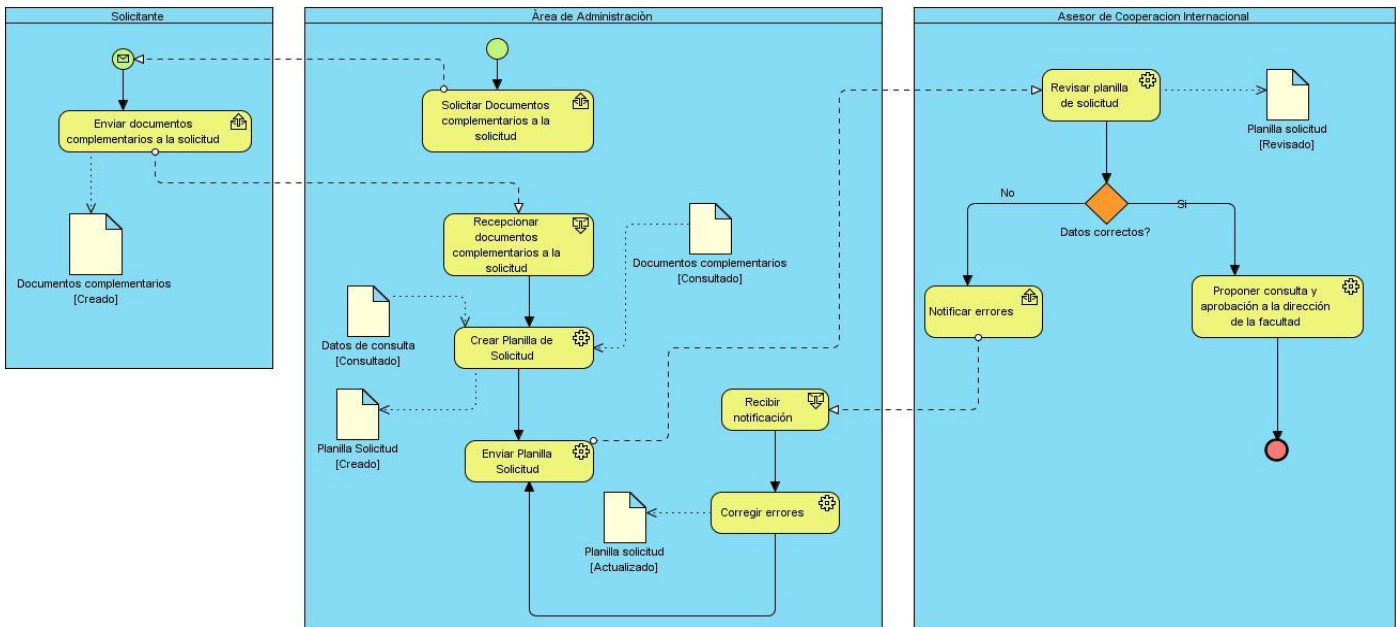


Figura 4 Descripción del proceso Crear planilla de solicitud de misión

2.4.3 Descripción del subproceso Aprobar solicitud de misión

Luego de que la Decana es informada de la existencia de una posible misión por parte del Asesor de Cooperación Internacional debe presentar dicha solicitud ante el Consejo de Dirección para proceder a la aprobación o no de la misma siendo éste uno de los procesos más importantes a desarrollar ya que durante el mismo se realiza la votación y se decide si la solicitud de misión presentada es aprobada. En este proceso tiene lugar también la selección de los entes que participarán en dicha votación y se definen de ellos los que vetan o no el voto realizado así como también las reglas de votación a tener en cuenta en el análisis de los resultados del sufragio. Con la realización de este proceso satisfactoriamente se define si el solicitante puede continuar el proceso de trámites de la solicitud a nivel de Rectoría. Independientemente del resultado de la votación realizada se deja constancia de la decisión tomada para tener en cuenta en próximos análisis.

Para obtener una descripción más detallada del flujo básico de actividades consultar el Anexo 6.

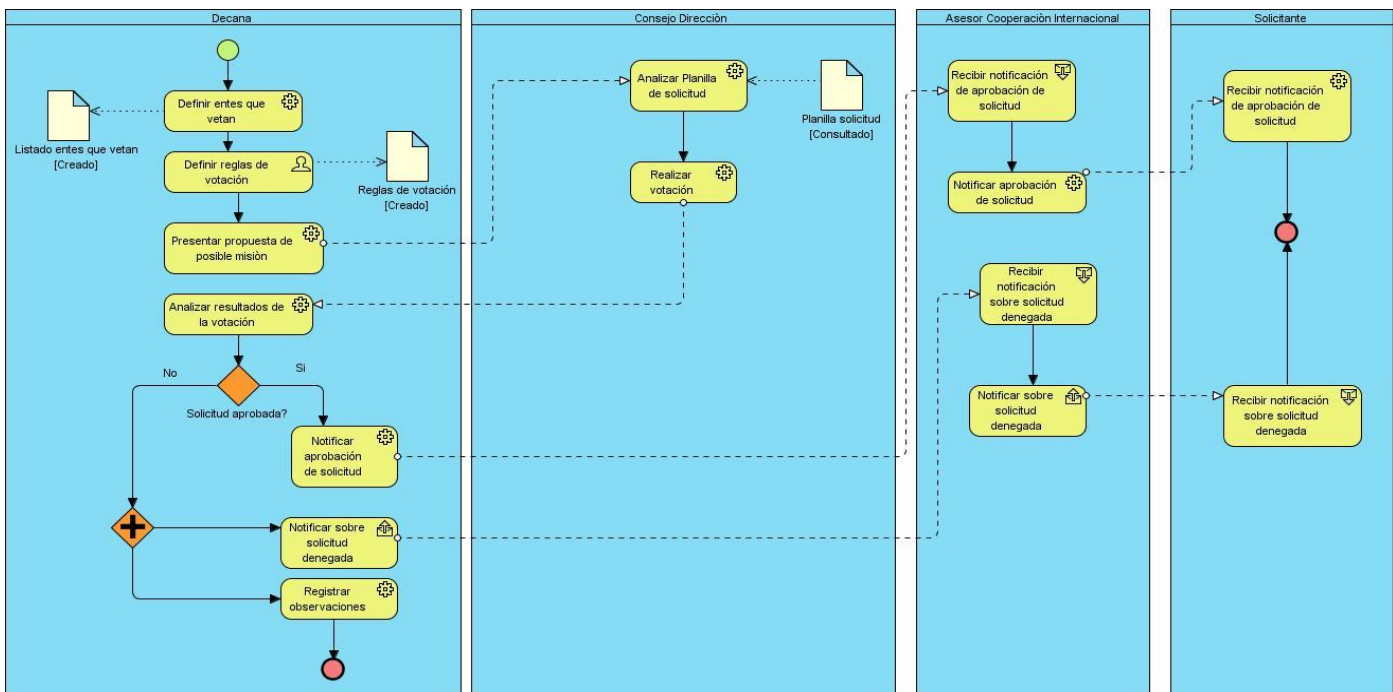


Figura 5 Descripción del subproceso Aprobar solicitud de misión

2.5 Validación de los procesos de negocio

Una vez culminada la identificación y descripción de los procesos de negocio se hace necesario validarlos. Esta validación tiene como propósito asegurar que cada proceso se halla descrito correctamente. Por esta razón se realizó un encuentro con el Asesor de Cooperación Internacional de la facultad quien revisó cada uno de los procesos descritos y determinó que la modelación correspondiente a cada uno de ellos coincidía con la realización de las actividades que se llevan a cabo en la Facultad 3 para tramitar la una solicitud de misión de cooperación internacional.

2.6 Modelo Conceptual

El Modelo conceptual constituye una representación de conceptos del mundo real. Explica los conceptos más significativos en el dominio del problema; es uno de los artefactos más importante a crear durante el análisis. Es un diagrama que ilustra una serie de relaciones entre ciertos factores que se cree impactan o conducen a una condición de interés. (23)

A continuación se representan las clases conceptuales del dominio del problema así como los conceptos del mundo real, no de los componentes de software.

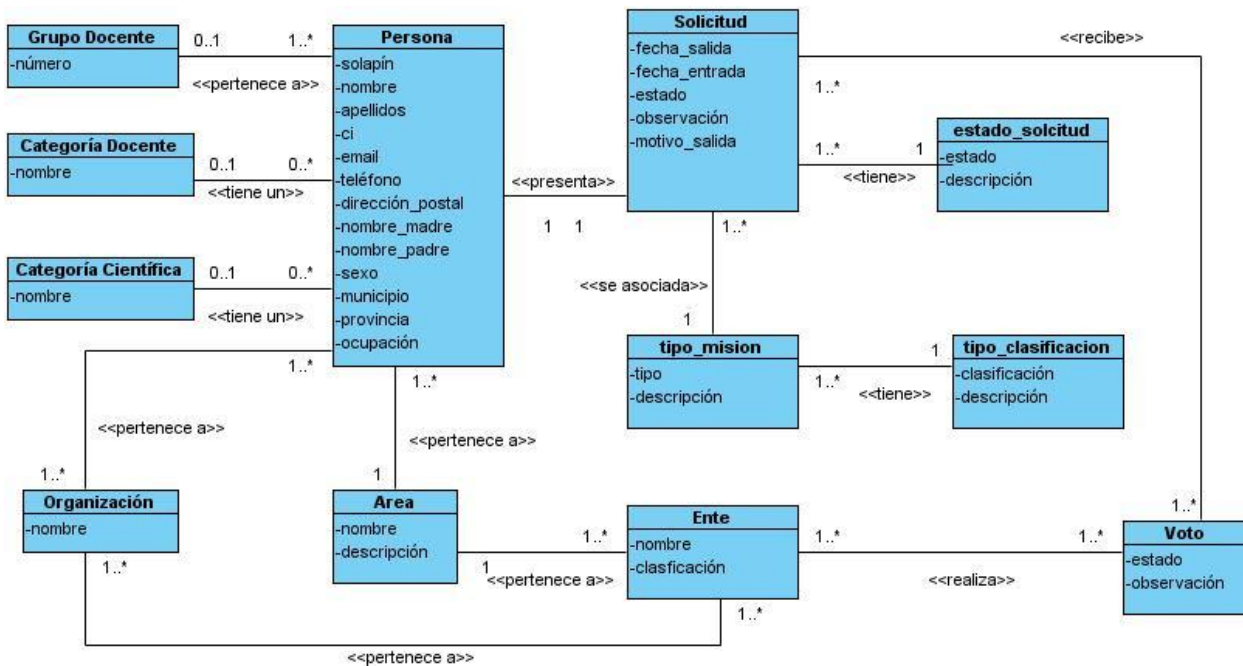


Figura 6 Modelo Conceptual

2. 7 Requisitos

Una vez desarrollados los epígrafes anteriores se puede concluir que el Modelamiento del Negocio brinda una visión general de los procesos que existen en un determinado Negocio, permitiendo establecer una comunicación entre el cliente y el equipo de desarrollo. Una vez llegado a este punto se hace necesario comenzar a definir lo que debe hacer el sistema y es por ello que se procede a identificar los requisitos que el mismo debe cumplir.

La IEEE⁶ Standar Glossary of Software Engineering Terminology define un requerimiento o requisito como: [1] “Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. [2] Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. [3] Una representación documentada de una condición o capacidad como en [1] o [2]”. (24)

⁶ Institute of Electrical and Electronics Engineers (IEEE). (Instituto de Ingenieros Eléctricos y Electrónicos), una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas.

2.8 Técnicas de Captura de Requisitos

Lograr una comunicación efectiva entre los usuarios y el equipo de proyecto con el objetivo de llegar a un entendimiento de lo que hay que hacer, es la clave del éxito en la producción de un software. Durante muchos años muchas aplicaciones han fallado porque existieron incongruencias entre lo que el usuario quería, lo que realmente necesitaba, lo que interpretaba cada miembro del equipo de proyecto y lo que realmente se obtiene. A fin de eliminar estos inconvenientes se realiza la captura de requisitos, actividad mediante la cual se extrae, de una fuente de información disponible, las necesidades que debe cubrir el sistema. El desarrollo de esta actividad puede resultar difícil, todo depende del entorno de trabajo en que se realice y de las personas que participan en la captura de los requisitos. Es por ello que se han desarrollado técnicas que permiten, de forma eficiente y precisa, realizar esta actividad.

A continuación se describen algunas de las técnicas de captura de requisitos existentes:

Introspección: consiste en tratar de imaginar qué le gustaría al usuario; tratando de ponerse en su lugar. Actualmente resulta poco efectivo ya que nadie puede ponerse en lugar de otro sin omitir una gran cantidad de detalles.

Investigación documental: consiste en el estudio de documentos escritos sobre un objeto determinado, es decir son todos aquellos documentos registrados en diferentes dispositivos físicos a los que se puede tener acceso en forma directa o indirecta para su consulta.

Entrevistas: Es el método clásico, se prepara la entrevista en el puesto de trabajo, para después aplicarla al cliente, que es la persona que más conoce del tema que se está indagando.

Tormenta de ideas: consiste en la realización de reuniones entre las partes interesadas en las que todos expresan sus ideas sobre el problema y su solución. La forma de llevarla a cabo es que cada participante diga su idea sin ser interrumpido por otro. Una vez concluida la reunión se recolectan las ideas sin duplicar las mismas.

Cuestionarios: se realizan a fin de obtener información acerca de actitudes, creencias, comportamiento y características de las personas claves en la organización donde se realizará el sistema de software. Debe ser simple, específico, no patrocinado, con precisión técnica y dirigida a quienes deben conocerlo.

Análisis de protocolos: consiste en pedirles a los usuarios potenciales que describan en voz alta las actividades que realizan dentro del sistema.

Observación: consiste en profundizar sobre lo que realmente se está haciendo manual o automáticamente en el sistema actual. La información que se obtiene a través de la observación incluye las actividades que realizan, los mensajes relacionados y las influencias recibidas.

Análisis del discurso: propone llevarse a cabo en el ambiente organizacional donde el software trabajará, es decir en el ambiente del usuario, y consiste en conversaciones en el que el orden de interacción se produce en tiempo real, de preguntas y respuestas rápidas e incluye el análisis de unidades del discurso desde los puntos de vista social y lingüístico.

Las técnicas de captura de requisitos empleadas fueron **Entrevistas, Tormenta de Ideas e Investigación Documental**. Se realizaron entrevistas a diferentes personas involucradas en el proceso de gestión de trámites de salida al exterior, principalmente al Asesor de Cooperación Internacional. Se realizaron tormentas de ideas a través de las cuales tuvieron lugar una serie de debates entre los miembros del equipo de trabajo en los que cada uno de ellos expuso sus ideas sobre el problema a resolver y la solución al mismo. Se realizó además una investigación de los documentos de cooperación internacional establecidos para realizar los trámites de salida al exterior a fin de conocer las normas establecidas para tramitar las solicitudes de los diferentes tipos de misiones definidos en la Universidad.

2.9 Identificación de requisitos

2.9.1 Requisitos Funcionales

Una vez que han sido aplicadas las técnicas de captura de requisitos descritas anteriormente se procede a su identificación, actividad en la cual se derivan los requerimientos del sistema de forma tal que quede definido cómo el sistema se ajusta mejor a las necesidades del negocio y cómo va a ser utilizado el mismo por los usuarios. Se identificaron un total de 65 requisitos funcionales los cuales se listan a continuación:

RF1 Enviar Solicitud Interna.

RF2 Llenar Solicitud.

RF3 Listar Solicitud.

Gestionar Solicitud

RF4 Adicionar Solicitud Interna desde el Inicio.

RF5 Adicionar Solicitud Interna Pendiente.

RF6 Adicionar Solicitud Externa.

RF7 Modificar Solicitud.

RF8 Eliminar Solicitud.

RF9 Revisar Solicitud.

RF10 Realizar Votación.

RF11 Aceptar Votación.

RF12 Analizar Estado de la Votación.

Configurar entes que vetan

RF13 Seleccionar entes que vetan siempre.

RF14 Editar entes que vetan siempre.

RF15 Seleccionar entes que vetan siempre según Área Administrativa.

RF16 Editar entes que vetan siempre según Área Administrativa.

RF17 Seleccionar entes que vetan siempre según Ocupación.

RF18 Editar entes que vetan siempre según Ocupación.

RF19 Seleccionar entes que vetan siempre según Membrecía.

RF20 Editar entes que vetan siempre según Membrecía.

Gestionar nomenclador Grupo

RF21 Adicionar nomenclador Grupo.

RF22 Modificar nomenclador Grupo.

RF23 Eliminar nomenclador Grupo.

Gestionar nomenclador Ocupación.

RF24 Adicionar nomenclador Ocupación.

RF25 Modificar nomenclador Ocupación.

RF26 Eliminar nomenclador Ocupación.

Gestionar nomenclador Categoría Docente.

RF27 Adicionar nomenclador Categoría Docente.

RF28 Modificar nomenclador Categoría Docente.

RF29 Eliminar nomenclador Categoría Docente.

Gestionar nomenclador Categoría Científica.

RF30 Adicionar nomenclador Categoría Científica.

RF31 Modificar nomenclador Categoría Científica.

RF32 Eliminar nomenclador Categoría Científica.

Gestionar nomenclador Área.

RF33 Adicionar nomenclador Área.

RF34 Modificar nomenclador Área.

RF35 Eliminar nomenclador Área.

Gestionar nomenclador Provincia.

RF36 Adicionar nomenclador Provincia.

RF37 Modificar nomenclador Provincia.

RF38 Eliminar nomenclador Provincia.

Gestionar nomenclador Municipio.

RF39 Adicionar nomenclador Municipio.

RF40 Modificar nomenclador Municipio.

RF41 Eliminar nomenclador Municipio.

Gestionar nomenclador País destino.

RF42 Adicionar nomenclador País destino.

RF43 Modificar nomenclador País destino.

RF44 Eliminar nomenclador País destino.

Gestionar nomenclador dinámico.

RF45 Adicionar nomenclador dinámico.

RF46 Modificar nomenclador dinámico.

RF47 Eliminar nomenclador dinámico.

RF48 Vincular nomenclador Municipio según Provincia

RF49 Crear nomenclador dinámico Clasificación según Tipo de Misión.

Gestionar nomenclador Estado Final.

RF50 Adicionar nomenclador Estado Final.

RF51 Modificar nomenclador Estado Final.

RF52 Eliminar nomenclador Estado Final.

Gestionar nomenclador Entes.

RF53 Adicionar nomenclador Entes.

RF54 Modificar nomenclador Entes.

RF55 Eliminar nomenclador Entes.

Gestionar nomenclador Membrecía.

RF56 Adicionar nomenclador Membrecía.

RF57 Modificar nomenclador Membrecía.

RF58 Eliminar nomenclador Membrecía.

Gestionar nomenclador Clasificación.

RF59 Adicionar nomenclador Clasificación.

RF60 Modificar nomenclador Clasificación.

RF61 Eliminar nomenclador Clasificación.

Gestionar nomenclador Tipo de Misión.

RF62 Adicionar nomenclador Tipo de Misión.

RF63 Modificar nomenclador Tipo de Misión.

RF64 Eliminar nomenclador Tipo de Misión.

RF65 Listar Valores del Nomenclador.

2.9.2 Requisitos No Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto, normalmente están vinculados a requisitos funcionales, es decir una vez se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. (25)

El módulo Gestión de Trámites para la Salida al Exterior al cual, a través el presente trabajo, se le realiza el análisis y diseño forma parte del Sistema Informatizado de Cooperación Internacional de la Facultad 3 que desarrolla el Grupo de Desarrollo de la facultad por lo que las propiedades o cualidades que debe presentar la aplicación web propuesta se acogen a las establecidas por dicho grupo. A continuación se presenta un resumen de algunos de los requisitos no funcionales fundamentales a tener en cuenta.

- Usabilidad

La aplicación desarrollada debe prestar facilidades de usabilidad que satisfagan las necesidades de todos los usuarios. Debe contar con un menú que les permita a los usuarios acceder a las principales funciones que son de su interés; así como brindarles la posibilidad de interactuar con los diferentes productos sin tener previa preparación, solo con los conocimientos necesarios del negocio. En este aspecto no se incluye la parte administrativa de las aplicaciones que si requerirán de una preparación para su manipulación. La resolución de la página se adaptará la resolución de pantalla en cada cliente.

- Fiabilidad

Disponibilidad: La aplicación deben ser capaz de estar operativa durante el mayor tiempo posible para brindar sus servicios a los usuarios ininterrumpidamente durante el tiempo que estos lo necesiten. El tiempo para el mantenimiento de la aplicación no debe exceder el 10% con respecto al tiempo que la aplicación debe estar disponible. El tiempo mínimo de disponibilidad ante posibles fallos será en dependencia de las especificaciones del sistema. La recuperación ante los fallos será en dependencia de las especificaciones del sistema.

- Eficiencia

Tiempo de respuesta por transacción: 1.09 segundos.

Rendimiento: 1000 transacciones por segundos, cantidad de datos que pueden ser transferidos.

Capacidad: 80 clientes que pueden estar conectados.

Utilización de recursos: 18 % memoria y 20 % en disco duro.

- Soporte

Las aplicaciones clientes deben funcionar sobre cualquier plataforma. Para la plataforma Windows, se recomienda utilizar la versión XP por la experiencia acumulada por los usuarios. Para la parte servidora se recomienda que corra sobre plataforma Linux. Debe ser programado en PHP 5.x o superior y con un gestor de base de datos MySQL 5.1 o superior. El sistema debe poseer una alta seguridad.

- Software

Los usuarios deben tener instalado como navegador el Mozilla Firefox 3.5, el Internet Explorer 6 o superior. Sistema Operativo Windows NT o Linux en las PC clientes. El servidor de aplicación debe tener instalado PHP 5.2.7 o superior, Servidor Web Apache 2.2.10 o superior. En la Base de Datos es necesario que esté instalada una versión de MySQL 5.1 o superior.

- Hardware

Cliente: La aplicación es desarrollada para que las PC clientes de los usuarios puedan hacer uso de la misma con el menor requerimiento posible. Estos requerimientos mínimos son:

Procesador Pentium IV o superior, 333 MHz mínimo pero recomendado 1.8 GHz, 40 GB o más de capacidad. Mínimo de memoria RAM 128 DIM, Adaptador de Red y conectividad.

Se recomienda la utilización de clientes ligeros para aprovechar las facilidades de mantenimiento que ofrece su uso.

Servidor: Debido a que la cantidad de datos manejados por este tipo de aplicaciones se requiere separar el servidor de aplicación y el de Base de Datos. Cada uno de ellos requiere de características de Hardware específicas.

Servidor de Aplicaciones:

CPU = 2 Dual Core AMD Opteron 64 bits a 2.2 GHz

RAM = 2 GB

Almacenamiento = 1 Disco SATA de 73 GB en RAID 1

Servidor de Base de Datos:

CPU = 2 Dual Core AMD Opteron 64 bits a 2.2 GHz

RAM = 4 GB

Almacenamiento = 2 Discos SATA de 73 GB en RAID 1

- **Restricciones de diseño**

La aplicación para gestionar los trámites de salida al exterior en la facultad incluye el manejo de varios asuntos que se enlazan entre sí, por lo que constantemente se necesita acceder a información común para realizar varios procesos, interviniendo en este caso la reutilización de dicha información de forma tal que los datos no aparezcan duplicados.

2.10 Especificación de requisitos

La especificación de requisitos es un factor principal para el buen funcionamiento del desarrollo de software. De acuerdo con el estándar IEEE 830, se considera que una especificación es de calidad cuando se puede decir que es correcta, no-ambigua, completa, consistente, ordenada por importancia y estabilidad, verificable, modificable y trazable.

En el Anexo 8 aparecen especificados algunos de los requisitos funcionales obtenidos. En la documentación adjunta al presente trabajo se pueden encontrar las especificaciones de cada uno de los requisitos funcionales identificados.

2.11 Técnicas de validación de requisitos

Además de las técnicas de captura de requisitos existen otras técnicas que permiten la validación de los mismos. Los requisitos, una vez definidos, deben de ser validados y para ello se aplican las técnicas de validación. Estas técnicas tienen como objetivo validar que la definición realizada de los requisitos coincide con lo que el cliente o el usuario final necesitan.

La aplicación de las técnicas de validación de requisitos seleccionadas en el desarrollo de la solución se evidencia durante el desarrollo del Capítulo 4 del presente trabajo. A continuación se describen algunas de estas técnicas.

Prototipo de interfaz

Durante la actividad de captura de requisitos, puede ocurrir que algunos requerimientos no estén demasiado claros o que no se esté muy seguro de haberlos entendido correctamente, todo lo cual puede llevar a un desarrollo no eficaz del sistema final. Para validar los requerimientos capturados, se diseñan prototipos de interfaz. Estos son simulaciones del posible producto, que luego son utilizados por el usuario final, permitiendo conseguir una importante retroalimentación en cuanto a si el sistema diseñado con base a los requerimientos capturados le permite al usuario realizar su trabajo de manera eficiente y efectiva.

La Revisión Técnica Formal (RTF)

Son reuniones del personal técnico (usuario final del sistema) con el objetivo de validar la especificación de requisitos. Su aplicación a los documentos de práctica permitirá detectar deficiencias, ambigüedades, omisiones, errores, etc., tanto de formato como de contenido. Suelen realizarla entre 3 y 5 personas y los especialistas funcionales deben ser independientes del equipo que ha realizado la especificación de requisitos.

Listas de chequeo

Son frecuentemente usadas en inspecciones o revisiones de artefactos generados en el proceso de producción de software; son listas de aspectos que deben ser completados o verificados.

Auditorías

Revisar la documentación con esta técnica, consiste en un chequeo de los resultados contra una (Listas de Chequeo) predefinida o definida a comienzos del proceso, es decir sólo una muestra es revisada.

Casos de Prueba

El objetivo del proceso de diseño de casos de prueba es crear un conjunto de casos de prueba que sean efectivos descubriendo defectos en los programas y muestren que el sistema satisface sus requerimientos.

Reviews o Walk-throughs

Esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar la consistencia de la documentación o información faltante.

2.12 Métricas de calidad de especificación de requisitos.

Para evaluar la calidad de especificación de requisitos existen un grupo de características que Pressman propone emplear para tal fin. Estas características se presentan en un conjunto de métricas enfocadas a evaluar la especificidad, corrección, completión, comprensión, capacidad de verificación, consistencia externa e interna, capacidad de logro, concisión, trazabilidad, capacidad de modificación, exactitud y capacidad de reutilización de un requisito.

Una métrica es un instrumento que cuantifica un criterio y persigue comprender mejor la calidad del producto, estimar la efectividad del proceso y mejorar la calidad del trabajo realizado. A continuación se describen las métricas que serán empleadas para evaluar la calidad especificación de requisitos que servirán de ayuda para valorar la completitud, consistencia y modificabilidad de una especificación.

Métrica de especificidad: esta métrica tiene como objetivo cuantificar la especificidad o falta de ambigüedad en la definición de los requisitos. Permite determinar si los requisitos están bien definidos, si no existen errores que afecten el diseño, que estén por orden de importancia, o sea establecer prioridades a la hora de desarrollarlos, además de determinar si están descritos detalladamente. (25)

Métrica de estabilidad: el objetivo de esta métrica es medir la estabilidad de los requisitos antes de pasar al siguiente flujo de trabajo. Se puede considerar que los requisitos son estables cuando no existen adiciones o supresiones en ellos que puedan implicar cambios en las funcionalidades principales de la aplicación. (26)

Promedio de requerimientos por necesidades: esta métrica verifica que las necesidades de los clientes se encuentren cubiertas por los requisitos especificados. (26)

2.13 Conclusiones

A través del desarrollo de este capítulo se inició el análisis de la propuesta de solución con el Modelo del Negocio donde se describen cada uno de los procesos identificados quedando reflejados en el Modelo de procesos la relaciones entre cada uno de ellos. Posterior al negocio se aplicaron varias técnicas que permitieron la captura de los requisitos funcionales del sistema, se especificaron los mismos y se propusieron otras técnicas a emplear para su validación así como un conjunto de métricas de calidad de especificación de requisitos.

CAPÍTULO 3: DISEÑO DEL SISTEMA

3.1 Introducción

En el presente capítulo se aborda el diseño de la propuesta de solución para satisfacer los requisitos especificados durante el análisis. Se expone el diagrama de componentes, el modelo de datos, el diagrama de clases y se realiza el modelado de cada una de ellas basado en el patrón arquitectónico Modelo Vista Controlador. Por último se proponen los patrones de diseño a emplear en la solución así como otros patrones que utiliza Joomla.

3.2 Arquitectura base

La arquitectura de software es la organización fundamental de un sistema, representada en sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución. Establece los fundamentos para que analistas, diseñadores, programadores, etc., trabajen en una línea común que permita alcanzar los objetivos del sistema, cubriendo todas las necesidades. Teniendo en cuenta que para realizar el análisis y el diseño del presente trabajo se adoptó el Modelo de Desarrollo definido por el proyecto ERP-Cuba no es objetivo crear una nueva arquitectura para el desarrollo de la aplicación si se cuenta con la arquitectura base de este proyecto la cual propone el desarrollo por componentes.

La arquitectura basada en componentes tiene como objetivo hacer un uso correcto de software reutilizable, para la construcción de aplicaciones de software mediante el ensamblaje de partes ya existentes. Un componente es un fragmento reemplazable de un sistema de software, una unidad de composición con interfaces especificadas contractualmente, que satisface una o varias funcionalidades dentro del contexto de una arquitectura bien definida y puede ser ensamblado con otros fragmentos por medio de una interfaz. Un componente puede contener múltiples objetos, clases y otros componentes.

(27) (9)

3.3 Patrón arquitectónico Modelo Vista Controlador

El Modelo-Vista-Controlador (MVC) es un patrón de arquitectura utilizado en sistemas web para separar los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos, permitiendo flexibilidad y facilidad a la hora de hacer futuros cambios. Es ampliamente utilizado en múltiples plataformas y lenguajes.

El **Modelo** contiene las funciones llamadas "lógica de negocio", o sea datos y reglas de negocio. Lleva un registro de las vistas y controladores del sistema. Cada acceso a datos se pone en su función individual porque, de esta forma, si se cambia de gestor de bases de datos este cambio sólo afecta a estas funciones, no al resto de la aplicación. Tener el modelo bien delimitado permite la existencia de varias aplicaciones que compartan el mismo modelo. (28)

La **Vista**, en una aplicación web, muestra la información del modelo al usuario. Tienen un registro de su controlador asociado (normalmente porque además lo instancia). (28)

El **Controlador** une la vista y el modelo. Por ejemplo, son las funciones que toman los valores de un formulario, consultan la base de datos a través del modelo y producen valores, que la vista tomará y convertirá en HTML. En resumen, gestiona las respuestas del sistema ante entradas del usuario. (28)

Entre las ventajas del estilo Modelo-Vista-Controlador están las siguientes:

Soporte de múltiples vistas: dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación web pueden utilizar el mismo modelo de objetos mostrado de maneras diferentes.

Adaptación al cambio: los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos dispositivos. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

Se hace uso de este patrón dadas las ventajas que brinda y las facilidades que ofrece a Joomla. Este CMS propone el uso de tres clases bases: JView, JModel y JController que responden a su arquitectura MVC.

La clase **JView** dispone del código que provee de datos dinámicos a la página y los métodos que permiten gestionar la presentación al usuario. La clase **JModel** se encarga del tratamiento de la información contenida en la base de datos, representa la información del mundo real que el sistema debe procesar. La clase **JController** se encarga de dirigir el flujo del control de la aplicación debido a mensajes externos como datos introducidos por el usuario u opciones del menú seleccionadas por él. (29)

3.3.1 El patrón MVC en los componentes Joomla

Un controlador en Joomla es una clase que extiende JController, con un método para cada tarea con la que el controlador tenga que tratar: mostrar un elemento, crear un elemento, borrar un elemento, etc. En estos métodos el controlador analiza la entrada del usuario y lleva a cabo los pasos necesarios para generar la respuesta (seleccionar el modelo o modelos necesarios y la vista a utilizar y asociarlos, hacer una redirección, generar un error).

Un modelo en Joomla es una clase que extiende JModel, con distintos métodos para realizar distintas acciones sobre los datos. No importa la forma que tomen estos datos o la forma en la que se trabaje con ellos.

Una vista en Joomla es una clase que extiende JView, con uno o más métodos para generar la página HTML, que muestra al usuario como respuesta. Cuando la respuesta es una página HTML, la vista suele hacer uso de plantillas.

3.4 Descripción y presentación del diagrama de componentes

El módulo Gestión de Trámites para la Salida al Exterior del Sistema Integral de Cooperación Internacional de la Facultad 3 está conformado por un conjunto de componentes, interrelacionados entre sí, en los que se encuentran agrupadas un grupo de funcionalidades que permiten tramitar una solicitud de cooperación internacional.

Con el objetivo de representar los componentes, sus interfaces y las relaciones de los componentes con las interfaces que utilizan se realiza el Diagrama de Componentes. Previamente se definieron estos componentes los cuales se describen a continuación.

Componente Configuración_Nomencladores: se encarga de adicionar, modificar, eliminar y listar los valores de los nomencladores.

Componente Configuración_Entes: en este componente se encuentran agrupadas una serie de funcionalidades referentes a la selección y edición de los entes, tanto los entes que vetan siempre, como los que vetan por área, membrecía y ocupación. Hace uso de algunos de los nomencladores configurados en el componente Configuración_Nomencladores.

Componente Votación: este componente agrupa las funcionalidades Realizar Votación, Aceptar Votación y Analizar Estado de la Votación. Hace uso de algunos de los nomencladores configurados en el componente Configuración Nomencladores así como también del componente Configuración_Entes y Solicitud.

Componente Solicitud: en este componente se encarga de Gestionar Solicitud, Enviar, Llenar, Listar y Revisar un grupo de solicitudes. Se comunica con los componentes Configuración_Nomencladores y Votación respectivamente.

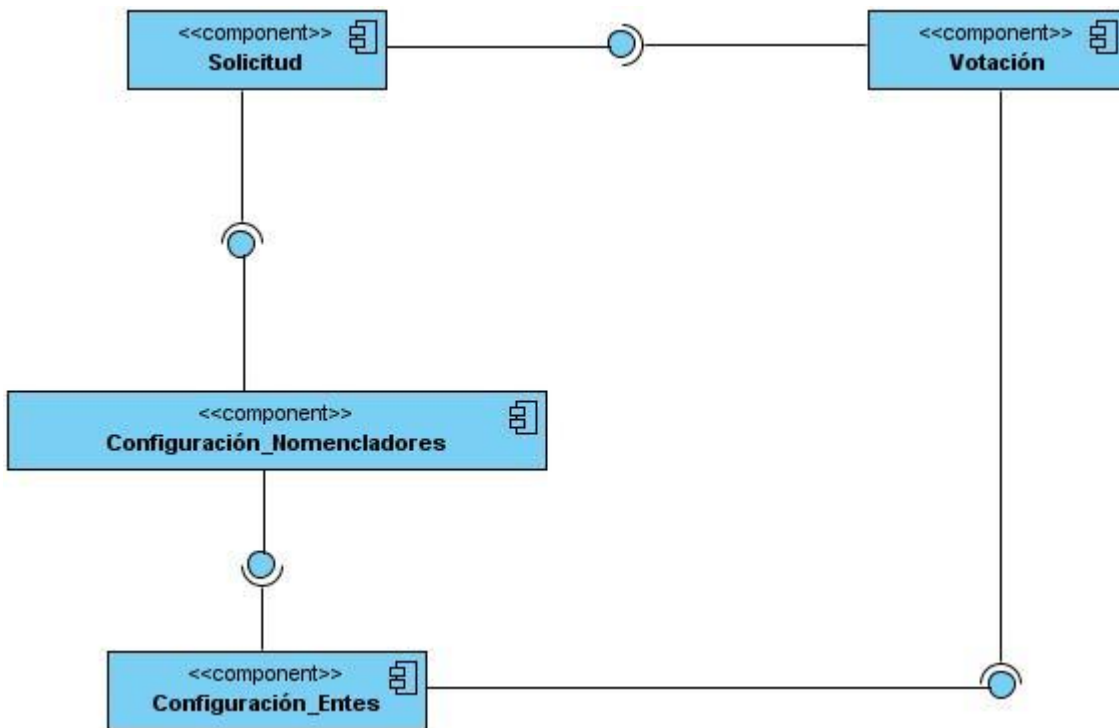


Figura 7 Diagrama de Componentes propuesto

3.5 Modelo de Datos

Un modelo de datos es una colección de conceptos bien definidos matemáticamente que ayudan a expresar las propiedades estáticas y dinámicas de una aplicación con un uso de datos intensivo. (30)

Para realizar la descripción de los datos contenidos en la base de datos (BD) se utilizará el modelo relacional. En el modelo de datos que se propone a continuación se encuentran representados los datos y relaciones entre éstos a través de una colección de tablas, en las cuales los renglones (tuplas) equivalen a

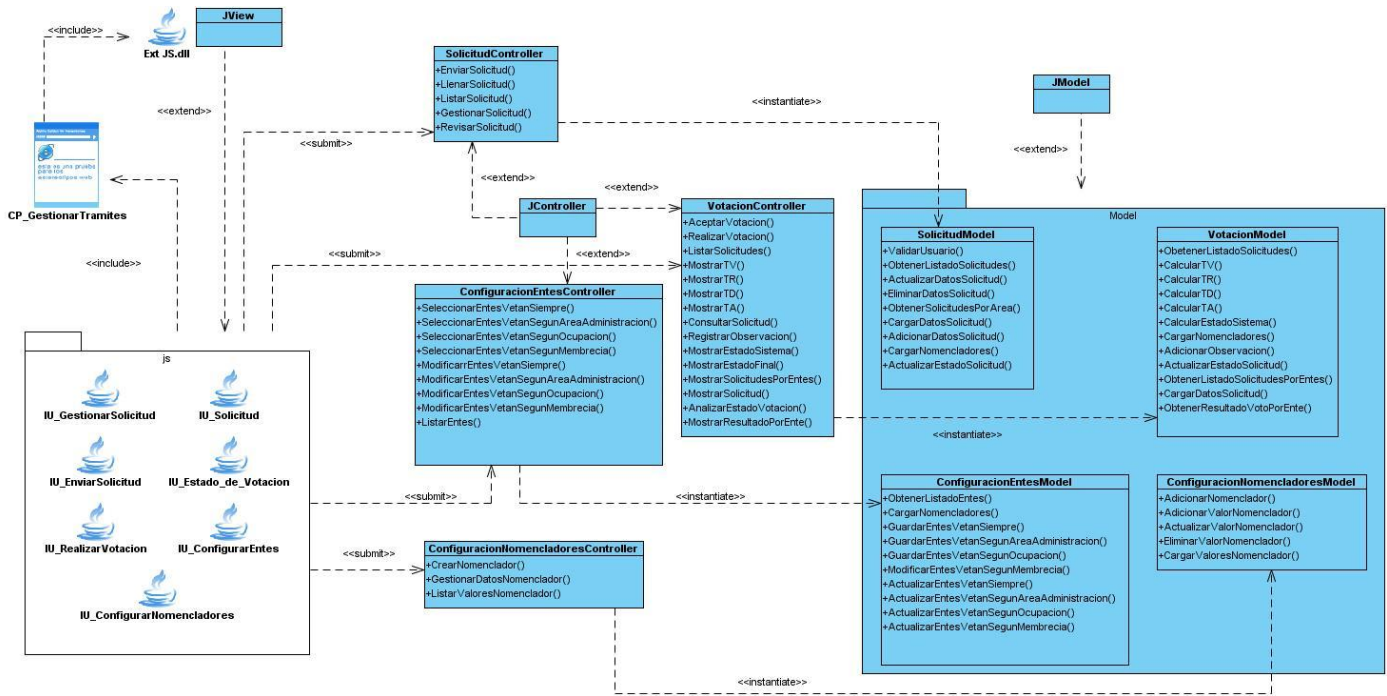


Figura 9 Diagrama de Clases del Diseño propuesto

3.6.1 Descripción del diagrama de clases del diseño propuesto

Tabla 1 Descripción del diseño de clases realizado

Clases	Descripción
Librería ExtJs (ExtJS.dll)	Contiene los componentes generados a través de la librería JavaScript ExtJs.
CP_GestionarTramites	Página encargada de visualizar, a través de los ficheros js que debe incluir, la aplicación y la información relacionada con la gestión de trámites de salida al exterior.
Clases js (IU_GestionarSolicitud.js, IU_Solicitud.js, IU_EnviarSolicitud.js, IU_Estado_de_Votacion.js, IU_RealizarVotacion.js, IU_ConfigurarEntes.js, IU_ConfigurarNomencladores.js, etc.)	Encargadas de generar de forma dinámica (que extiende de la clase base JView de Joomla) utilizando la librería ExtJs los componentes que manejen la información en la vista necesaria para el trabajo en la aplicación. Debe enviar y recibir los datos de la controladora utilizando tecnología AJAX.
SolicitudController	Clase controladora responsable de configurar los destinos donde se deben ejecutar las diferentes funcionalidades referentes a las

	solicitudes, según las peticiones del usuario. Extiende de la clase base de Joomla JController.
ConfiguracionEntesController	Clase controladora responsable de configurar los destinos donde se deben ejecutar las diferentes funcionalidades referentes a la configuración de los entes, según las peticiones del usuario. Extiende de la clase base de Joomla JController.
ConfiguracionNomencladoresController	Clase controladora responsable de configurar los destinos donde se deben ejecutar las diferentes funcionalidades referentes a configuración de los nomencladores a emplear, según las peticiones del usuario. Extiende de la clase base de Joomla JController.
VotacionController	Clase controladora responsable de configurar los destinos donde se deben ejecutar las diferentes funcionalidades referentes al proceso de votación, según las peticiones del usuario. Extiende de la clase base de Joomla JController.
Paquete Model	Encargado de manejar los datos persistentes dentro del componente a través de las clases del modelo SolicitudModel, VotacionModel, ConfiguracionEntesModel y ConfiguracionNomencladoresModel las cuales extienden de la clase base de Joomla JModel.
JView	Clase que dispone del código que provee de datos dinámicos a la página y los métodos que permiten gestionar la presentación al usuario.
JController	Clase que se encarga de dirigir el flujo del control de la aplicación.
JModel	Clase que se encarga del tratamiento de la información contenida en la base de datos, representa la información del mundo real que el sistema debe procesar.

3.7 Patrones de diseño

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan.

(31)

Para la realización del diseño se tuvieron en cuenta un conjunto de patrones que permiten solucionar problemas de diseño que posteriormente puedan presentarse lo cual facilitaría el trabajo durante el desarrollo de la aplicación. El presente trabajo no persigue como objetivo un estudio minucioso de los

patrones de diseño por lo que para obtener una descripción completa de patrones de diseño, se recomienda consultar el libro *Design Patterns: Elements of Reusable Object-Oriented Software*, escrito por the Gang of Four el cual es considerado como una guía y referencia a los patrones de diseño de software. A continuación se exponen algunos patrones que se encuentran presentes en la solución.

Se emplearon los patrones de asignación de responsabilidades GRASP (General Responsibility Assignment Software Patterns), entre ellos, los patrones Experto, Controlador, Bajo acoplamiento y Alta Cohesión.

El patrón **Experto** plantea que se debe asignar la responsabilidad al experto en información, que es la clase que cuenta con la información necesaria para cumplir la responsabilidad.

El patrón **Controlador** establece que se debe asignar la responsabilidad del manejo de los eventos de un sistema a una clase controladora.

El patrón **Bajo Acoplamiento** persigue tener las clases lo menos ligadas entre sí que se pueda. De esta manera, en caso de producirse una modificación en alguna de ellas, se tiene la mínima repercusión posible en el resto de las clases. Con este patrón se potencia la reutilización y se disminuye la dependencia entre las clases.

La cohesión es una medida de cuán relacionadas y enfocadas estén las responsabilidades de una clase. La aplicación del patrón **Alta Cohesión** facilita un diseño donde las clases tienen responsabilidades estrechamente relacionadas y no realizan un trabajo enorme.

Joomla, por su parte, utiliza numerosos patrones de diseño presentes en su marco de trabajo, entre ellos, y utilizados en la solución se encuentran el patrón **Factory** y **Singleton** respectivamente. A continuación se describe brevemente su empleo.

El patrón de la fábrica (**Factory**) es un patrón que se utiliza para construir y devolver objetos. Es utilizado en los casos donde diferentes clases, por lo general derivan de una clase abstracta, son instanciadas dependiendo de los parámetros. Joomla proporciona la clase estática JFactory, que implementa el patrón de la fábrica. Esta clase es importante porque permite acceder de forma sencilla a instancias de los objetos globales. Este ejemplo muestra se puede acceder a algunos de los objetos globales utilizando JFactory. (32)

```
$db = &JFactory::getDBO();  
$user = &JFactory::getUser();  
$document = &JFactory::getDocument();
```

El patrón **Singleton** (singleton pattern) se utiliza para permitir la creación de sólo un único objeto de una clase específica. Se consigue haciendo el constructor privado o protegido y usando un método estático para crear instancias de la clase. Muchas de las clases de Joomla utilizan un modelo pseudo-singleton que permite crear instancias de objetos y de acceso. Para lograrlo se utiliza un método estático llamado `getInstance()`; en algunos casos `JFactory` actúa como una interfaz a través de este método. (29)

3.8 Conclusiones

Con la realización del presente capítulo se expuso el diseño de la solución propuesta para satisfacer los requisitos. En el mismo se describió la arquitectura que rige el diseño y el patrón arquitectónico MVC. Se construyó el diagrama de componentes que muestra las relaciones entre cada uno de los componentes identificados donde se agruparon las funcionalidades que debe brindar el sistema. Se obtuvieron además otros artefactos relativos al diseño definidos en el modelo de desarrollo adoptado. Por último se describieron un conjunto de patrones de diseño a emplear en la solución.

CAPÍTULO 4: VALIDACIÓN DEL SISTEMA

4.1 Introducción

En el presente capítulo se procede a validar la solución propuesta. Se proponen un conjunto de prototipos de interfaces de usuario que permiten validar los requisitos identificados durante el análisis. Se aplican métricas para medir la calidad de especificación de requisitos así como otras que permiten validar el diseño de clases realizado.

4.2 Aplicación de las técnicas para la validación de requisitos

Una vez que los requisitos hayan sido descritos se hace necesario validar los mismos. La validación de requisitos es una actividad muy importante, pues un levantamiento de requisitos con errores que no se detecten a tiempo, además de no conducir a resultados inesperados provoca costos excesivos y gran pérdida de tiempo. (24)

En el Anexo 9 del presente trabajo se encuentra reflejada el Acta de aprobación de requisitos que fue elaborada y firmada por el Asesor de Cooperación Internacional de la facultad y la analista principal del proyecto luego de haber realizado una revisión de cada una de las especificaciones de requisitos capturados.

4.2.1 Revisión Técnica Formal

La validación de requisitos examina la especificación para asegurar que todos los requisitos de software se han establecido de manera precisa; que se han detectado las inconsistencias, omisiones y errores y que éstos han sido corregidos, y que los productos de trabajo cumplen con los estándares establecidos para el proceso, proyecto y producto. (12)

El mecanismo primario para la validación de requisitos es la Revisión Técnica Formal, técnica que fue aplicada una vez concluida la especificación de éstos, se realizó la reunión de revisión con el analista principal del proyecto SICI donde quedaron aprobadas las especificaciones descritas.

4.3.2 Prototipos

Durante la actividad de extracción de requerimientos, puede ocurrir que algunos requerimientos no estén demasiado claros o que no se esté muy seguro de haber entendido correctamente los requerimientos obtenidos hasta el momento, todo lo cual puede llevar a un desarrollo no eficaz del sistema final. Entonces, para validar los requerimientos hallados, además de realizar una revisión de la especificación de los mismos, se construyen prototipos. Los prototipos son simulaciones del posible producto, que luego son utilizados por el usuario final, permitiendo conseguir una importante retroalimentación en cuanto a si el sistema diseñado con base a los requerimientos recolectados le permite al usuario realizar su trabajo de manera eficiente y efectiva.

A continuación se ilustran y se ofrece una breve descripción de los prototipos de interfaz de usuario obtenidos.

Descripción: Prototipo de Interfaz de Usuario (IU) que permite el envío interno de una solicitud.



El prototipo muestra una ventana con el título "Enviar solicitud". Dentro de la ventana, hay dos campos de texto etiquetados "Correo Electrónico". El segundo campo tiene un enlace "más" a su derecha. En la parte inferior derecha de la ventana, hay un botón "Enviar".

Figura 10 IU_Enviar Solicitud Interna

Descripción: Prototipo de IU a través del cual se ingresan los datos de una solicitud de misión de cooperación internacional. Contiene solamente los datos personales del solicitante y responde a la funcionalidad de Llenar Solicitud.

The screenshot shows a web application window titled "Solicitud" with the following fields and controls:

- Top Bar:** "Número de solapín" (input), "Cargar" (button), "Limpiar datos" (button), and a "FOTO" placeholder.
- Personal Information:** "Nombre" (input), "Apellidos" (input), "Carné Identidad" (input), "Teléfono" (input), "Nombre de la madre" (input), "Nombre del padre" (input), "Ocupación" (dropdown), "Sexo" (input), "Grupo" (dropdown), "Provincia" (dropdown), "Municipio" (dropdown), and "Dirección particular" (input).
- Membership and Requester:** "Membrecía" with checkboxes for FEU, PCC, UJC, and CTC; "Solicitado por:" with checkboxes for "Area Administración" and "Area de Vinculación".
- Professional Information:** "Categoría Docente" (dropdown), "Categoría Científica" (dropdown), "Area Administración" (dropdown), and "Area de Vinculación" (dropdown).
- Identification:** "Carné Identidad 1" (input with image icon) and "Carné Identidad 2" (input with image icon).
- Bottom Bar:** "Limpiar Campos" (button), "Aceptar" (button), "Aplicar" (button), and "Cancelar" (button).

Figura 11 IU_Llenar Solicitud

Descripción: Prototipo de IU a través del cual se gestiona una solicitud. Permite listar las solicitudes, adicionar, modificar y eliminar solicitudes. A través de esta interfaz se pueden consultar los datos completos de una solicitud y realizar la revisión de la misma.

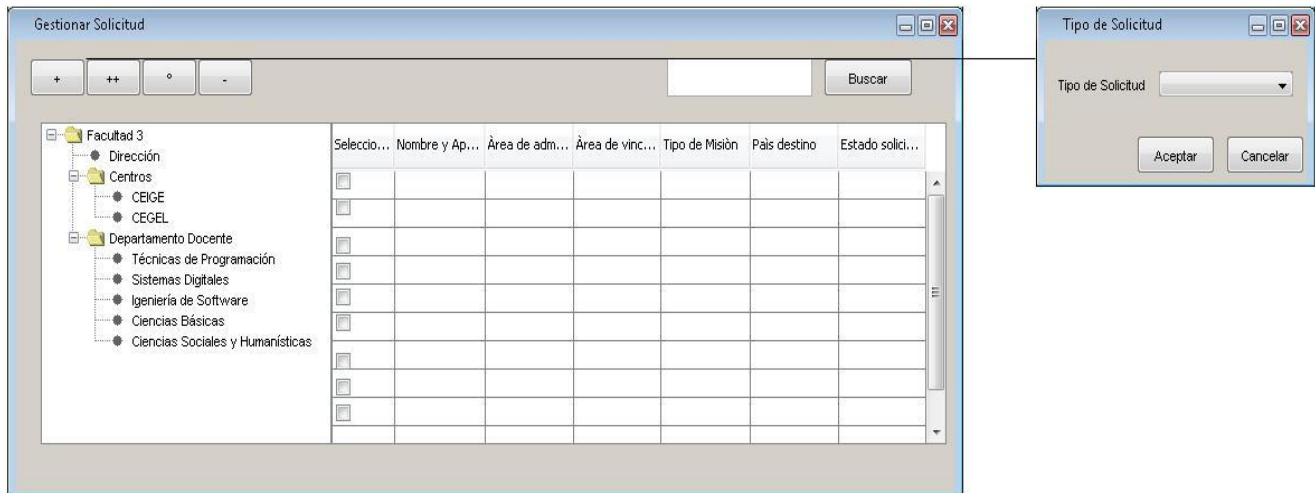


Figura 12 IU_Gestionar Solicitud

Figura 13 IU_Solicitud

Descripción: Prototipo de IU a través del cual se realiza la aceptación de la votación. A través de ella se pueden registrar observaciones sobre una solicitud o una decisión tomada sobre ésta. Desde la misma interfaz se puede acceder a la planilla de solicitud y consultar los datos recogidos en ella.

The 'Aceptar Votación' window features a search bar at the top left with a 'Buscar' button. Below it is a table with the following columns: 'Nombre y Apelli...', 'Área que Soli...', 'País Destino', 'Tipo de Misión', 'Clasificación', 'Fecha de Sal...', 'TV', 'TR', 'TA', 'TD', 'Estado Sistema', and 'Estado Final'. The table contains several empty rows. To the left of the table is a vertical list of checkboxes. Below the table is a large text area labeled 'Observaciones'. At the bottom right, there are four buttons: 'Cancelar', 'Aceptar', 'Aplicar', and 'Detalles'.

The 'Solicitud' window is a detailed form for entering personal and mission data. It includes fields for 'Nombre', 'Apellidos', 'Carné Identidad', 'Correo electrónico', 'Teléfono', 'Nombre de la madre', 'Nombre del padre', and 'Ocupación'. There are also dropdown menus for 'Sexo', 'Grupo', 'Provincia', 'Municipio', 'Categoría Docente', 'Categoría Científica', 'Area Administración', and 'Area de Vinculación'. Checkboxes are provided for 'Membrecía' (FEU, PCC, UJC, CTC) and 'Solicitado por:'. Two fields for 'Carné Identidad' (1 and 2) include image upload icons. At the bottom, there are dropdowns for 'País Destino', 'Tipo de misión', 'Fecha de Salida', 'Fecha de Regreso', 'Clasificación', 'Estado Sistema', and 'Estado Final'. A 'Motivo de Salida' text area and an 'Observaciones' text area are also present. The bottom right corner contains 'Aceptar', 'Aplicar', and 'Cancelar' buttons.

Figura 14 IU_Aceptar Votación

Descripción: Prototipo de interfaz de usuario que muestra el estado de la votación. A través de esta interfaz se pueden ver los datos de los solicitantes por los cuales se realiza la votación, además permite analizar por cada una de las solicitudes existentes el estado final de la misma y el resultado del voto emitido por cada una de los entes que intervienen en la votación.

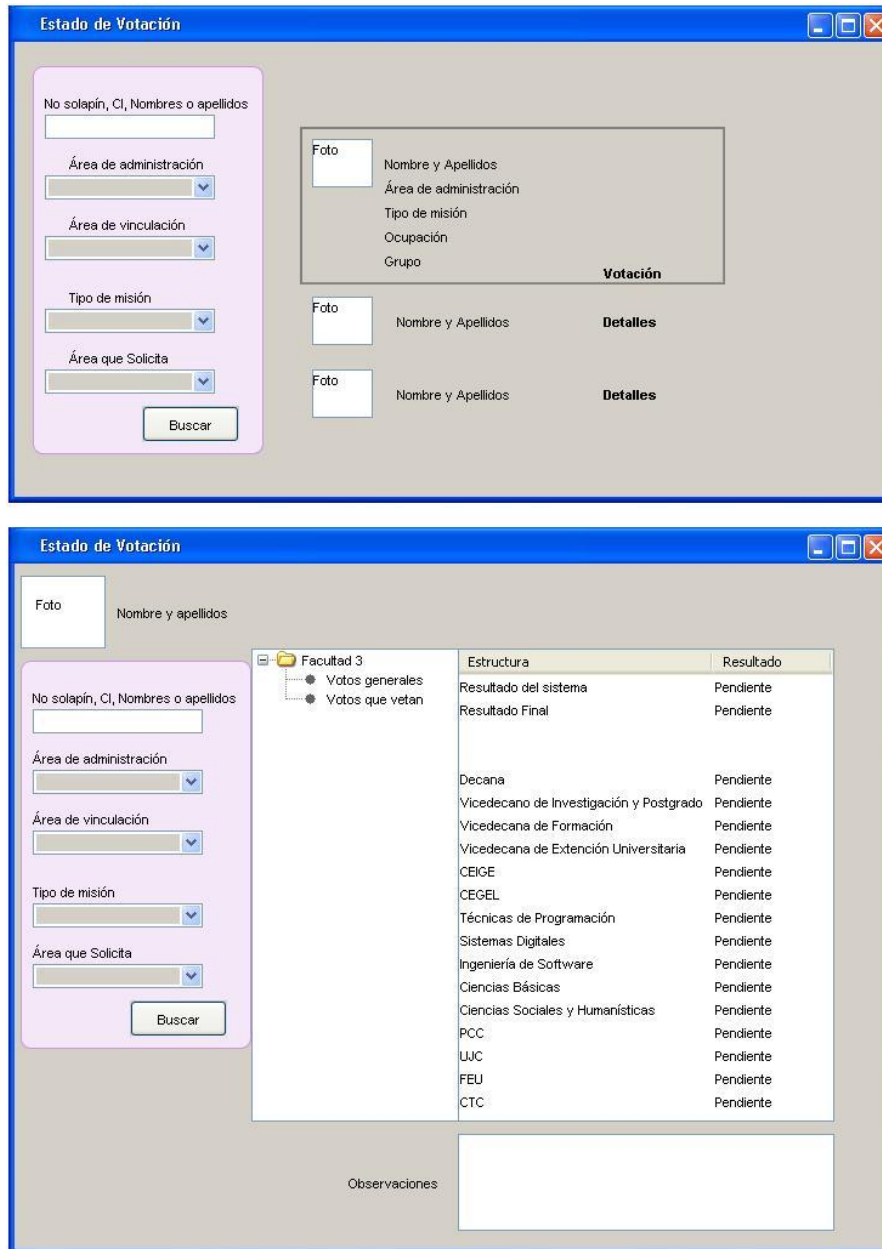


Figura 15 IU_Analizar Estado de la Votación

Descripción: Prototipo de interfaz de usuario a través del cual el ente realiza la votación por una solicitud determinada.

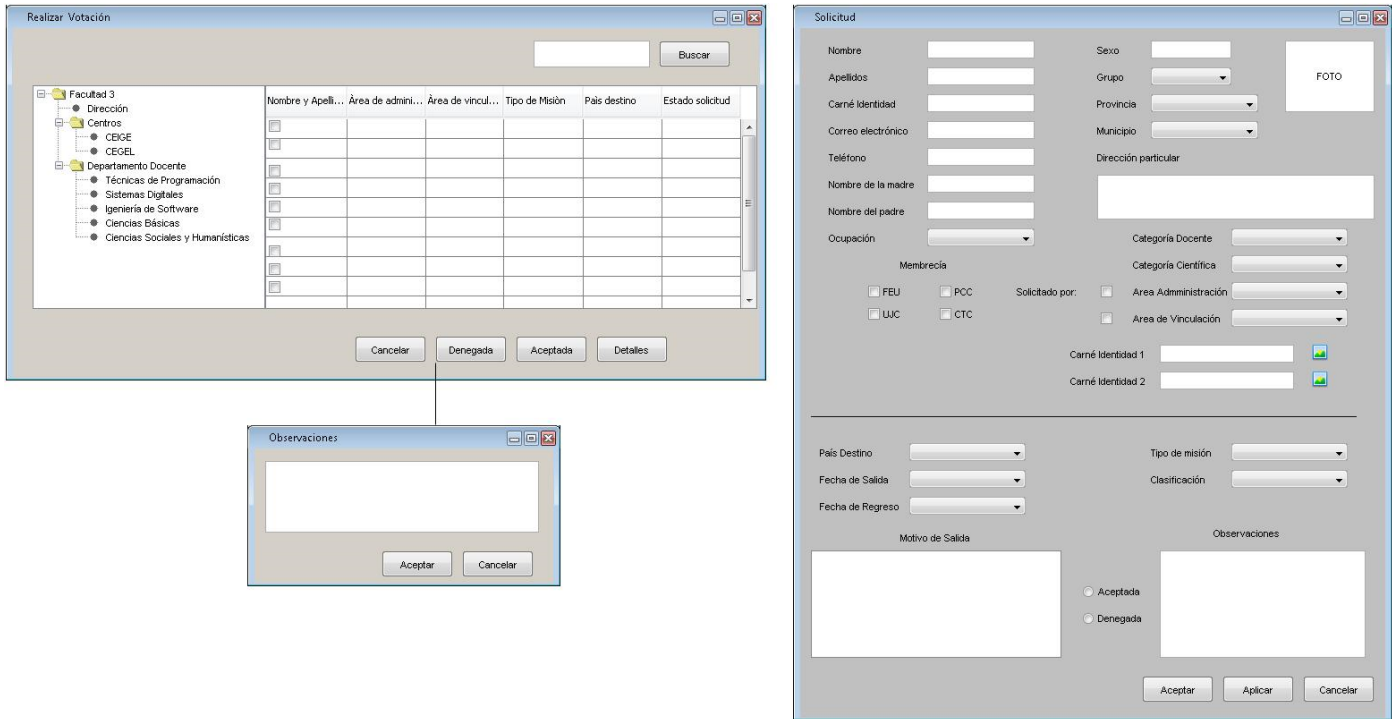


Figura 16 IU_Realizar Votación

Descripción: Prototipo de interfaz de usuario a través del cual se seleccionan y editan cada uno de los entes que vetan siempre, los que vetan según el área de administración a la que pertenecen, a la ocupación que presentan y de acuerdo a su membresía.



Figura 17 IU_Entes que Vetan

Descripción: Prototipo de interfaz de usuario a través del cual se realiza la vinculación de los municipios a la provincia. Permite definir para cada una de las provincias los municipios que tiene asignados.

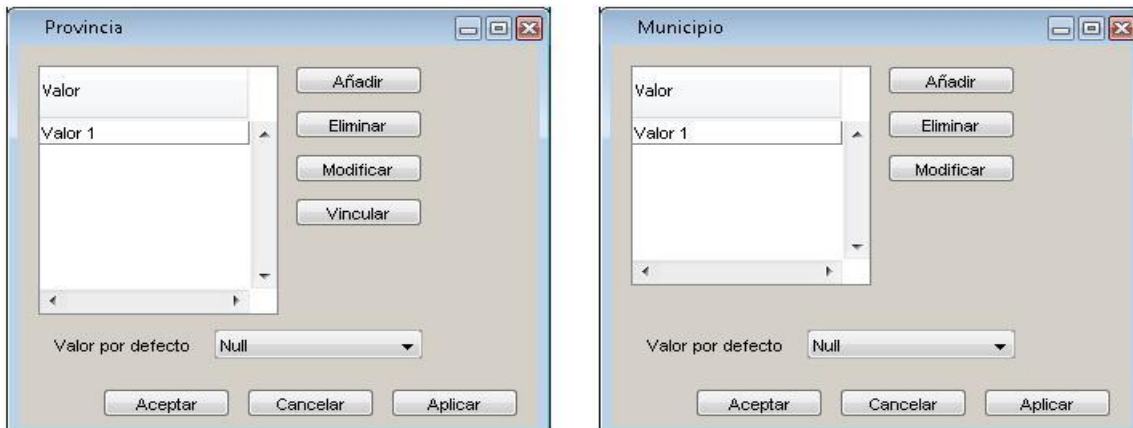


Figura 18 IU_ Vincular nomenclador Municipio según Provincia

Descripción: Prototipo de IU por medio de la cual se configuran cada uno de los nomencladores. A continuación se presenta la interfaz de configuración del nomenclador Tipo de Misión, donde a través de la ventana Tipo de Misión se adicionan, modifican o eliminan los valores del nomenclador. Comparten la

misma interfaz el resto de los nomencladores que aparecen relacionados en ella con la excepción que el nomenclador Provincia tienen agregada la opción Vincular para permitir agrupar los municipios según la provincia a la cual pertenecen.

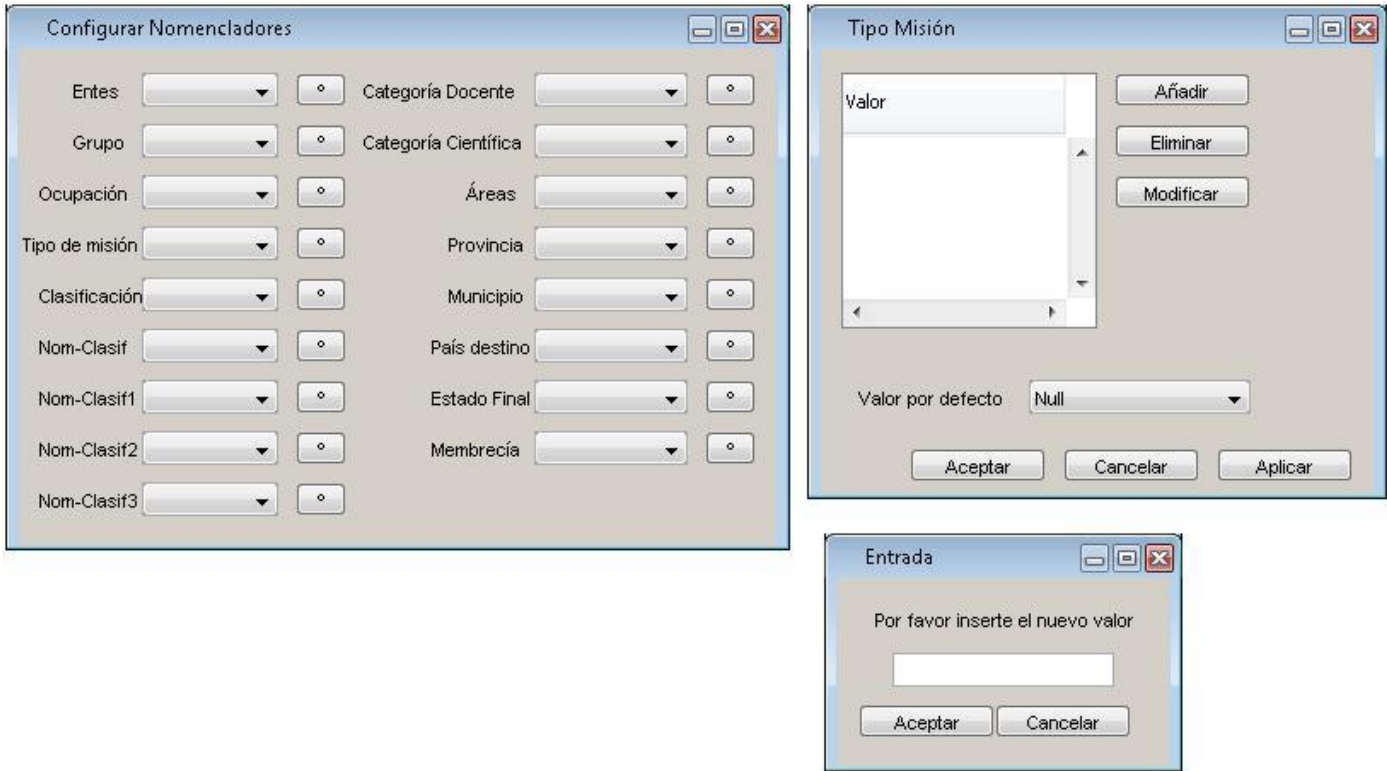


Figura 19 IU_Configurar Nomencladores

Descripción: Prototipo de IU a través del cual se crean nuevos nomencladores en relación al vínculo entre un tipo de misión y las clasificaciones que ésta pueda tener.

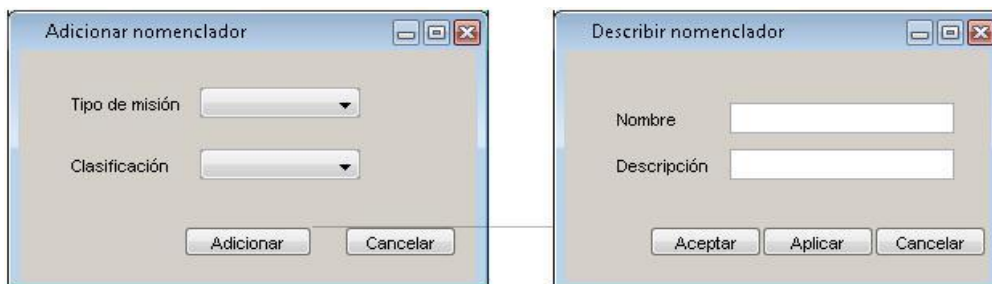


Figura 20 IU_Crear nomenclador dinámico

4.3 Aplicación de las métricas de calidad de la especificación de requisitos

A continuación se aplican métricas que permiten obtener valores cuantitativos para medir la calidad de las especificaciones de requisitos anteriormente realizadas.

Tabla 2 Tabla auxiliar para evaluar la calidad de la especificación de requisitos

Variables	Descripción	Valor
TR	Total de requisitos evaluados	65
NUI	Número de requisitos para los que todos los revisores tuvieron una misma interpretación	63
CR	Cantidad de requisitos cambiados (insertados, modificados y eliminados)	4
CN	Cantidad de necesidades de los clientes.	9
CNR	Cantidad de necesidades sin requerimientos.	0

Tabla 3 Aplicación de las métricas de calidad de especificación de requisitos

No	Métrica	Fórmula	Propiedad	Valor
1	Especificidad	$Q = \text{NUI} / \text{TR} * 100$	No ambigüedad	96,9 %
2	Estabilidad	$\text{ETR} = ((\text{TR}-\text{CR})/\text{TR})*100$	Estabilidad	93,8%
3	Promedio de requisitos por necesidades	$\text{PNR} = \text{TR}/\text{CN}$	Corrección	7,22

4.3.1 Resultados de las métricas aplicadas.

Una vez aplicadas las métricas de la calidad de la especificación de los requisitos se puede observar que existe un alto índice de especificidad debido a que posee un 96,9 % y mientras más cerca de 100 esté el valor de Q mayor es la consistencia de la interpretación de los revisores para cada requisito y menor la ambigüedad en la especificación de los mismos. Para la estabilidad el valor más óptimo es el más cercano a 100, para el caso de ETR se obtuvo un 93,8 % demostrándose que los requisitos son estables y por tanto es confiable trabajar el análisis y diseño sobre ellos. Con el valor obtenido en el promedio se

evidencia que no se quedaron necesidades sin requisitos. Los resultados obtenidos con la aplicación de las métricas fueron satisfactorios, demostrando la calidad de la especificación de los requisitos.

4.4 Evaluación del Modelo de Diseño

Las métricas de diseño a nivel de componentes se concentran en las características internas de los componentes del software e incluyen medidas de la cohesión, acoplamiento y complejidad del subsistema. Estas tres medidas pueden ayudar al desarrollador de software a juzgar la calidad de un diseño a nivel de los componentes.

Entre los tipos de Métricas que se utilizan para evaluar el diseño se encuentran:

Tamaño operacional de clase (TOC): está dado por el número de métodos asignados a una clase.

Relaciones entre clases (RC): está dado por el número de relaciones de uso de una clase con otras.

Profundidad de herencia (PH): está dado por la profundidad en la herencia de las clases heredadas de un nodo padre.

Número de descendientes (ND): está dado por la cantidad de clases que heredan de un padre.

Número de operaciones redefinidas para una clase hija (NOR): está dado por la cantidad de operaciones redefinidas en cada clase hija.

Para evaluar el diseño se emplearán solamente las métricas Tamaño operacional de clase (TOC) y Relaciones entre clases (RC) ya que no existen casos de herencia lo cual descarta la aplicación del resto de las métricas anteriormente expuestas. La primera mide la calidad de acuerdo a los atributos Responsabilidad, Complejidad de implementación y Reutilización de las clases mientras que la segunda brinda una medida de las relaciones de uso existentes entre cada clase.

4.5 Resultados de la evaluación de las métricas empleadas

Los cálculos realizados durante la aplicación de estas métricas pueden consultarse en el libro de Excel **Métricas para evaluar el diseño** adjunto al presente documento.

Aplicación de la Métrica TOC

Para la evaluación del diseño a través de esta métrica se tuvieron en cuenta la cantidad de métodos de cada una de las clases Controladoras y Modelos reflejadas en el diagrama de clases del diseño presentado.

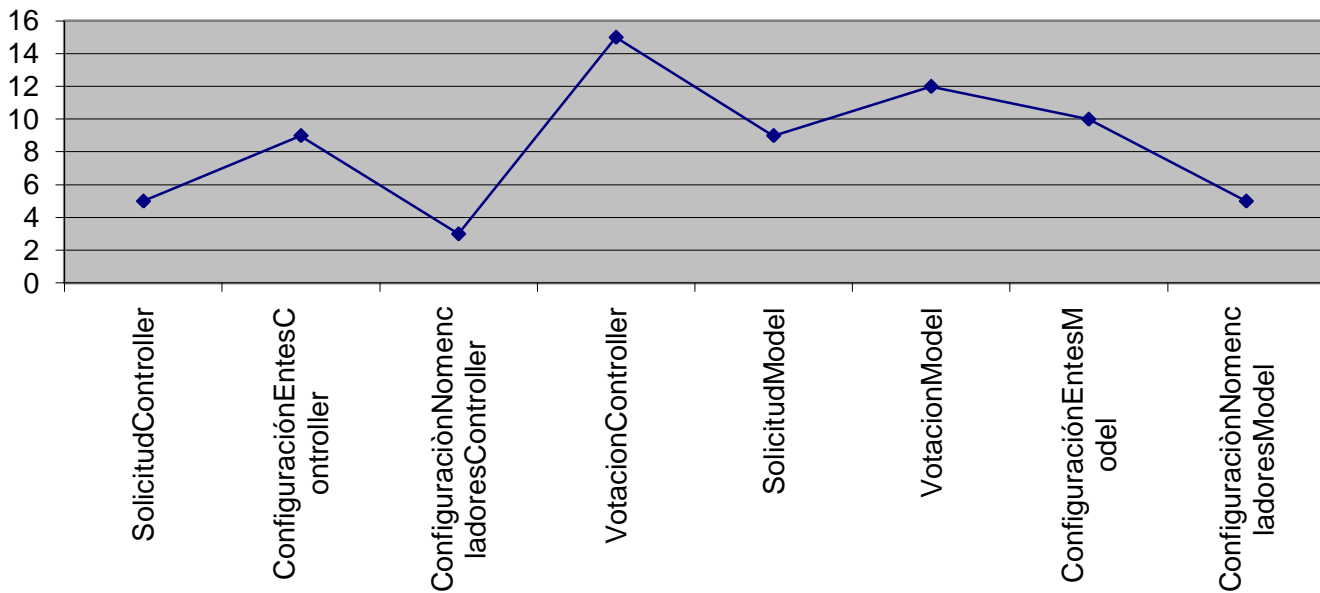


Figura 21 Gráfico Cantidad de procedimientos por clases

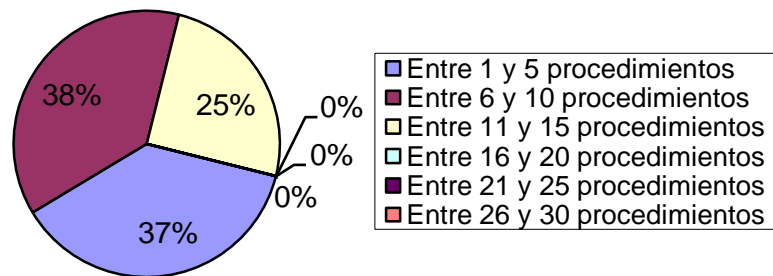


Figura 22 Gráfico de los resultados generales de acuerdo a la cantidad de procedimientos

Valoración

El análisis de los resultados obtenidos luego de la aplicación de la métrica TOC arroja que para un total de 8 clases evaluadas se realizaron 68 operaciones, obteniéndose un promedio de operaciones por clases de 8,5. La mayoría de las clases analizadas son de tamaño medio lo cual aumenta la reutilización de las clases e implica una baja responsabilidad de las mismas así como baja complejidad de las pruebas.

Aplicación de la Métrica RC

Para la evaluación del diseño a través de esta métrica se tuvieron en cuenta la cantidad de relaciones de uso de cada una de las clases Controladoras y Modelos reflejadas en el diagrama de clases del diseño presentado.

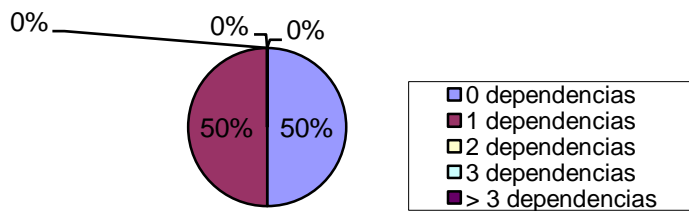


Figura 23 Gráfico de dependencias entre las clases

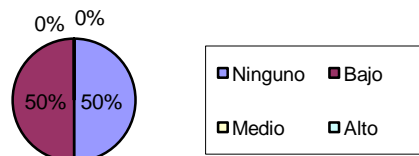


Figura 24 Gráfico de Acoplamiento entre clases

Valoración

Durante la evaluación de la métrica RC los resultados obtenidos demostraron que el 50% de las clases poseen menos de 2 dependencias entre estas para un promedio de 0,5 asociaciones de uso. Los índices de acoplamiento entre las clases son bajos lo cual influye en que en caso de producirse una modificación en alguna de ellas se tiene la mínima repercusión posible en el resto de las clases potenciando la reutilización de las clases.

4.6 Conclusiones

En este capítulo se expuso el diseño de la solución propuesta para satisfacer los requerimientos. En el mismo se mostró la arquitectura base que rige el diseño, los componentes y las funcionalidades agrupadas en ellos, el modelo de datos y el diagrama de clases del diseño. Se propuso el uso de patrones de diseño GRASP y otros que ofrece Joomla en busca de hacer más fácil la reutilización del diseño de clases. Se aplicaron métricas y técnicas para la validación de los requisitos además de realizar la validación del diseño de clases a través de métricas orientadas a clases las cuales arrojaron valores positivos.

CONCLUSIONES

- ✓ Se evidenció la necesidad de informatizar las actividades referentes a la Gestión de Trámites para la Salida al Exterior que se realiza en la Facultad 3.
- ✓ El modelo de desarrollo adoptado permitió la realización del trabajo de forma organizada y la obtención de los artefactos donde quedaron registradas cada una de las actividades desarrolladas.
- ✓ A través de la identificación de los procesos de negocio y la definición de los requisitos funcionales se realizó la descripción de una aplicación que satisface las necesidades del cliente referentes a los procesos Presentar propuesta de solicitud de misión, Crear planilla de solicitud de misión y Aprobar solicitud de misión que se desarrollan en la Facultad 3 como parte de la Gestión de Trámites para la Salida al Exterior.
- ✓ Se realizó el diseño del sistema utilizando patrones arquitectónicos y de diseño los cuales posibilitarán un mejor funcionamiento del sistema propuesto.
- ✓ Se evaluó el análisis y diseño del sistema aplicando técnicas y métricas que demostraron que los requisitos funcionales y el diseño de las clases no presentan dificultades y se determinó que con el trabajo realizado se puede proceder a la implementación de la solución propuesta.

RECOMENDACIONES

Después de haber sido realizadas las conclusiones del presente trabajo se recomienda:

- ✓ Se recomienda dar seguimiento al Análisis y Diseño realizado en el presente trabajo e iniciar las actividades de implementación que permitan poner en funcionamiento una aplicación web para la Gestión de Trámites para la Salida al Exterior en la Facultad 3.
- ✓ Una vez desarrollada la aplicación se aplique en las diferentes facultades de la Universidad.

REFERENCIAS BIBLIOGRAFICAS

1. **Rojas, Suset Fernández y Fonseca, Eduardo Luis Piña.** *Análisis y Diseño del Sistema Informatizado de Cooperación Internacional.Subsistema de Cooperación.* La Habana : s.n., 2008.
2. **Tellez, Raynel Batista.** *Manual de Cooperación Internacional de la Facultad 3.* La Habana : s.n., 2011.
3. **Ayala, Reisser Mejias y Ruiz, Keytia Quintero.** *Sistema Informatizado de Cooperación Internacional. Análisis y Diseño del Subsistema de Trámites.* La Habana : s.n., 2008.
4. **Software, Departamento de Ingeniería de.** Entorno Virtual de Aprendizaje. *Conferencia: FT Análisis y Diseño.Extensiones Web.* [En línea] [Citado el: 10 de Marzo de 2011.] <http://eva.uci.cu>.
5. **Valladares, Tania Teresa Laureiro y Rodríguez, Lázaro Amed Aguilar.** *Análisis y Diseño de la solución informática para el subsistema de Caja, del sistema de gestión empresarial Cedrux.* La Habana : s.n., 2009.
6. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** El Proceso Unificado de Desarrollo de Software. *Entorno Virtual de Aprendizaje.* [En línea] [Citado el: 14 de Marzo de 2011.] <http://eva.uci.cu>.
7. Metodologías de desarrollo de software. *Entorno Virtual de Aprendizaje.* [En línea] [Citado el: 20 de Marzo de 2011.] <http://eva.uci.cu>.
8. **Acuña, Karenny Brito.** *SELECCION DE METODOLOGIAS DE DESARROLLO PARA APLICACIONES WEB EN LA FACULTAD DE INFROMATICA DE LA UNIVERSIDAD DE CIENFUEGOS.* Cienfuegos : s.n., 2009.
9. **ERP-Cuba, Proyecto.** *Modelo de Desarrollo Orientado a Componentes del proyecto ERP-Cuba.* 2008.
10. **Stephen A. White, IBM Corporation.** Introduction to BPMN.
11. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** El Lenguaje Unificado de Modelado.Manual de Referencia. [En línea] 2000. [Citado el: 22 de Marzo de 2011.] <http://eva.uci.cu>.
12. **Pressman, Roger S.** Ingeniería de Software.Un enfoque práctico 6ta Edición. *Entorno Virtual de Aprendizaje.* [En línea] [Citado el: 22 de Marzo de 2011.] <http://eva.uci.cu>.
13. **International, V.P. 2006.** Introducción a los sistemas y herramientas CASE. 2006.
14. **GARCIA, X.C and J. M. ALFONSO.** Introducción a los Sistemas de Gestión de Contenidos. 2004.
15. *The Features, Mission, and Principles of the Drupal Project.* . 2008.
16. Sitio Oficial de Joomla! [En línea] 2011. [Citado el: 25 de Marzo de 2011.] <http://www.joomla.org>.
17. Conferencia 1: Introducción a la asignatura de sistemas de bases de datos. *Entorno Virtual de Aprendizaje.* [En línea] 2010. [Citado el: 28 de Marzo de 2011.]

18. **Gallego, José Antonio Vázquez.** Desarrollo Web con PHP y MySQL. 2003. Anaya Multimedia, ISBN 8441515255.
19. Documentacion del Servidor HTTP Apache 2.0. [En línea] 2009. [Citado el: 28 de Marzo de 2011.] <http://httpd.apache.org/docs/2.0/es/>.
20. Lenguaje HTML. [En línea] [Citado el: 28 de Marzo de 2011.] <http://www.desarrolloweb.com/articulos/711.php>.
21. **Southwell, Michael.** Pro PHP Security. 2005.
22. Fase de Inicio. Disciplina de Modelamiento del negocio. *Entorno Virtual de Aprendizaje*. [En línea] [Citado el: 8 de Abril de 2011.] <http://eva.uci.cu>.
23. Conferencia Introducción a la Disciplina Análisis y Diseño. *Entorno Virtual de Aprendizaje*. [En línea] [Citado el: 18 de Abril de 2011.] <http://eva.uci.cu>.
24. Conferencia Introducción a la Disciplina de Requisitos. *Entorno Virtual de Aprendizaje*. [En línea] [Citado el: 20 de Abril de 2011.] http://eva.uci.cu/file.php/102/Curso_2010-2011/Clases/Semana_06/Conf_7/Materiales_complementarios/Introduccion_a_la_Disciplina_de_Requisitos.pdf.
25. **Monzón, Antonio.** Calidad de la Especificación: ¿Se pueden medir los Requisitos?.
26. **Cobas, Leydis Castellanos.** *Modelado de Negocio y Requerimientos de los procesos de recepción y control de los Documentos Complementarios al Despacho Comercial*. La Habana : s.n., 2010.
27. **CEDRUX.** *Documento Especificación de la Arquitectura del Sistema e Integración ERP Cuba*. 2009.
28. **Ismarai Núñez Viltres, Abdiel Matos Nieto.** *Sistema de apoyo al proceso de enseñanza-aprendizaje desde juegos didácticos*. Ciudad de la Habana : s.n., 2009.
29. Joomlaos.net. *Framework Joomla!* [En línea] [Citado el: 6 de Mayo de 2011.] <http://www.joomlaos.net/>
30. **González, Dra. Anaisa Hernández.** Un método para el diseño de la base de datos a partir del modelo orientado a objetos.
31. **García, Joaquín.** Patrones de Diseño. [En línea] [Citado el: 23 de Mayo de 2011.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.
32. Joomlaos.net. [En línea] [Citado el: 23 de Mayo de 2011.] <http://www.joomlaos.net/framework-joomla-15x/106-el-patron-de-la-fabrica-factory>.
33. **Fisteus, Jesús Arias.** *Modelado de procesos de negocio. Aplicación en entornos móviles*. 2002.

