

Universidad de las Ciencias Informáticas
Facultad 3



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

Título: Subsistema para la gestión de los procesos de trámites de salidas y visitas, hacia y desde el exterior versión 2.0, en la Facultad 3.

Autores: Alfredo Álvarez Nápoles

Tutor: Ing. Cristian Fernández López

La Habana, junio de 2012

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 _____.

Alfredo Álvarez Nápoles

Ing. Cristian Fernández López

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

A mi mamá por estar en todo momento, por ser mi mejor amiga, mi consejera, mi apoyo incondicional, la mujer que lo daría todo por mi felicidad.

A mi papá por ser el motor impulsor de mis estudios.

A mi hermanita, la niñita de mis ojos por su cariño incondicional

A mis hermanos con los cuales siento la responsabilidad de estar a la altura de lo que ellos esperan.

A mis tíos por ser conmigo excepcionales.

A mis queridos abuelos que siempre han estado presentes para darme su apoyo y su amor.

A mi primo Manolito (pepe) por toda su ternura y amistad.

A mi amigo y hermano Ángel que siempre ha sabido darme el consejo preciso y me ha apoyado en los momentos difíciles.

A ANA, a la cual debo, por su amor, quien soy en mi presente. Siempre saca lo mejor de mí, ella ha sido mi todo y su ayuda en la realización de este trabajo ha sido fundamental.

A mis amigos, 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 Rodney, Yasiel, Gustavo, Jarsey, Rafael y a mi otro compañero de apto Yisel por estar siempre para todo. A mi familia, por el apoyo incondicional durante estos 5 años...

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

A Rainel por todo el apoyo brindado a lo largo del desarrollo de este trabajo.

A mis compañeros del basket en especial Nani y Piri los cuales han sido grandes compañeros...

DEDICATORIA

Quiero dedicar el presente trabajo, por el valor de su resultado, a mi padre. El cual no puede estar conmigo en este momento tan especial de mi vida, ya que siempre fue un guía para seguir en mi vida, el primero en exigirme y encaminarme profesionalmente. Para esa persona tan especial que no puede ver lo que ha crecido su hijo, dedico este trabajo.

Resumen

Con el objetivo de incrementar el impacto de la Universidad de las Ciencias Informáticas en la informatización del país, incrementar las exportaciones y fortalecer la formación pedagógica y científica de 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 y visita desde y hacia el 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 cooperación internacional. No se aprovechan al máximo las tecnologías existentes, afectándose la calidad de las mismas. De ahí la necesidad de emplear nuevas variantes que agilicen el flujo de actividades y que a su vez garanticen su calidad, eficiencia y rapidez.

Por tanto, el objetivo principal del presente trabajo se centra en la realización del Subsistema para la gestión de los procesos de trámites de salidas y visitas, hacia y desde el exterior, en La Facultad 3, que gestione adecuadamente las actividades necesarias para tramitar una solicitud de misión de Cooperación Internacional.

El presente documento recoge un estudio sobre otros sistemas de trámites existentes en el ámbito nacional y a su vez dentro de la universidad, además se describen las características de las herramientas usadas para el desarrollo de la propuesta de solución. Se especifican las funcionalidades que brinda el sistema a sus clientes, además se realiza el diseño, implementación y prueba de la aplicación web propuesta.

Palabras Claves

Trámites, misiones de Cooperación Internacional, procesos, gestión, aplicación web.

ÍNDICE

INTRODUCCIÓN	10
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	15
1.1 INTRODUCCIÓN	15
1.2 Marco conceptual	15
1.3 Sistemas de gestión de procesos de cooperación internacional	17
1.4 Metodología, Herramientas y tecnologías utilizadas	19
1.4.1 Aplicación web	19
1.4.2 Metodología de desarrollo	20
1.4.3 Lenguaje de modelado	22
1.4.4 Herramientas de Modelado	22
1.4.5 Visual Paradigm for UML	22
1.4.6 Gestor de Base de Datos	22
1.4.7 IDE de desarrollo	24
1.4.8 Tecnologías Web.	24
1.5 Conclusiones	29
CAPÍTULO 2: ANÁLIS Y DISEÑO DEL SUBSISTEMA	30
2.1 Introducción	30
2.2 Lista de reserva del producto	30
2.3 Historia de usuarios (HU)	35
2.4 Plan de Iteraciones	37
2.5 Patrones usados en el diseño	39
2.6 Tarjetas crc	44

2.7 Modelo de datos	46
2.8 Conclusiones	48
CAPÍTULO 3: IMPLMENTACIÓN Y PRUEBA	49
3.1 Introducción	49
3.2 Tareas de ingeniería	49
3.3 Diagrama de componentes	51
3.4 Diagrama de Despliegue	51
3.5 Pruebas	52
3.5.1 Principios de la prueba	52
3.6 Validación de la implementación.	55
3.6.1 Evaluación de la ejecución de las pruebas Unitarias.	55
3.6.2 Evaluación de la ejecución de las pruebas de aceptación.	59
3.6.3 Acta de Aceptación	61
3.7 Conclusiones	61
CONCLUSIONES	62
RECOMENDACIONES	63
BIBLIOGRAFÍA	64

ÍNDICE DE TABLAS

Tabla 1. Lista de reserva del producto.	30
Tabla 2. Requisitos no funcionales.	33
Tabla 3. Historia de usuario Gestionar solicitud interna.....	35
Tabla 4. Plan de iteraciones.....	38
Tabla 5. Tarjeta crc módulo 1 Nomencladores.....	44
Tabla 6. Tarea de Ingeniería 1 de la Historia de Usuario Gestionar Solicitud Interna.....	49
Tabla 7. Tarea de Ingeniería 2 de la Historia de Usuario Gestionar Solicitud Interna.....	50
Tabla 8. Tarea de Ingeniería 3 de la Historia de Usuario Gestionar Solicitud Interna.....	50
Tabla 12. Gráfica de resultados por iteraciones	60
Tabla 13. Prueba de aceptación de la H_U Realizar votación.....	61

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Clase PersonaExternaController	40
Ilustración 2. Clase PersonaExternaGestor.....	41
Ilustración 3. Clases PersonaExternaController, PersonaExternaGestor y Persona.	42
Ilustración 4. Método buscarTodosPorIDs.....	43
Ilustración 5. Clase SigeciController.....	44
Ilustración 6. Modelo de datos	47
Ilustración 7. Diagrama de componente.....	51
Ilustración 8. Diagrama de Despliegue.....	52
Ilustración 9. Prueba de caja blanca del método costoscreateAction	57
Ilustración 10. Grafo de Flujo Complejidad Ciclomática	58

INTRODUCCIÓN

La evolución de los programas y las políticas de cooperación internacional desde 1950 solo puede ser explicada en el contexto de la compleja trama de relaciones entre estados que constituyen la escena política mundial, referencia sin la cual no es posible comprender sus características fundamentales. La cooperación internacional para el desarrollo es una dimensión más de las relaciones internacionales. Así lo confirma la rápida revisión de los intereses que han animado a la comunidad internacional a poner en práctica programas de cooperación entre naciones. A los factores históricos y de política exterior se suma una pluralidad de intereses económicos, comerciales, geopolíticos, culturales y humanitarios, cuya peculiar combinación diferencia cada programa nacional de cooperación.

Cuba es uno de los países en vías de desarrollo en los que se ha tenido que desarrollar una política de colaboración internacional, en todas las esferas socio-económicas, de forma activa, responsable y basada en el respeto a los principios de la nación. La educación superior es una de las ramas socio-económicas importantes del país ya que muchas de las universidades producen productos para el mercado exterior que ayudan en lo económico al país, existen también intercambios de conocimientos, solicitudes de becas, cursos de posgrado y pregrado que son de gran ayuda para fomentar las relaciones sociales y culturales de las naciones implicadas. La cooperación internacional ha pasado a desempeñar un papel muy importante en estos procesos, y dentro de las acciones que implica, la relacionada con las actividades de gestión de proyectos de colaboración internacional (GPCI).

Los sistemas informáticos son una de las vías principales para la solución de los problemas que acogen al mundo actual, sumergiéndose así esta era en el conocimiento de la informática, la telecomunicación y la tecnología. La cooperación internacional también se acoge a las nuevas tendencias tecnológicas por lo que surgen sistemas informáticos para la gestión de trámites de cooperación internacional a toda escala y en diversos países. Cuba no queda exento de esta novedad y como prueba de ello se pone de manifiesto sistemas informáticos como el del Ministerio de la Industria Sideromecánica (SIME) o el del Ministerio de Educación Superior (MES).

La Universidad de las Ciencias Informáticas (UCI) tiene como una de sus principales misiones producir software y servicios informáticos a partir de la vinculación estudio – trabajo como modelo de formación. En la sede se efectúan procesos de cooperación internacional para la gestión de trámites de entrada y salida

al exterior, a nivel central y extendiéndose por las diversas facultades, aumentando el vínculo internacional y favoreciendo la reputación de la Universidad de las Ciencias Informáticas en el mundo.

En la facultad 3 los procesos de cooperación internacional son llevados a cabo en la actualidad de forma manual dificultando la rapidez de trabajo y la eficiencia ya que al realizarse de forma manual se corre el riesgo de la pérdida de información, todo esto aun después de las investigaciones de las autoras de los trabajos de diplomas titulados Subsistema para la gestión de los procesos de trámites de salidas y visitas, hacia y desde el exterior en la Facultad 3 y Análisis y diseño del módulo Gestión de Trámites para la Salida al Exterior del Sistema Integral de Cooperación Internacional de la Facultad 3. En su versión inicial el subsistema de gestión de procesos de trámites de salidas y visitas, hacia y desde el exterior no está actualizado tecnológicamente para integrarse con la plataforma de aplicaciones de la facultad 3 (SO3), no se gestiona el plan de actividades por lo que lo que no se conoce el estado por el que está pasando el proceso en su debido momento, el módulo de solicitudes se encuentra desactualizado funcionalmente ya que después de un análisis profundo de la primera versión muchas de las funcionalidades que se desarrollan en este módulo no cumplen con las reglas del negocio de acuerdo a lo pedido por el cliente de cooperación internacional, por lo que se necesita realizar una nueva captura de funcionalidades para que los procesos de gestión de trámites de cooperación internacional se puedan gestionar de la manera que exigen las reglas de negocio y por todo lo antes mencionado dificulta la adecuada gestión de la información de cooperación internacional en la facultad 3.

Por todo lo antes planteados se define el siguiente **problema a resolver**:

¿Cómo mejorar la gestión de la información de los procesos de trámites de entrada y salida desde y hacia el exterior de la facultad 3?

Se asume como **objeto de estudio**: Proceso de gestión de la información sobre trámites migratorios.

Campo de acción: Proceso tramitación de visitas hacia y desde el exterior en la facultad 3.

Objetivo General: Desarrollar la segunda versión del subsistema de gestión de los procesos de trámites de salidas y visitas, hacia y desde el exterior para mejorar la gestión de su información en la Facultad 3.

Objetivos específicos:

- Realizar un estudio del marco teórico referente a los sistemas de la gestión de los procesos de trámites de salidas y visitas.

- Elaborar una propuesta para desarrollar un sistema de gestión de los procesos de trámites de salidas y visitas, hacia y desde el exterior versión 2.0, en la Facultad 3.
- Validar la propuesta de solución mediante la realización de pruebas que evalúen el sistema (Pruebas de caja blanca y pruebas de aceptación).

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

- Análisis conceptual de los objetivos del área de relaciones internacionales, proceso, cultura de gestión y toma de decisiones.
- Caracterización de la situación actual del objeto de estudio en la Universidad de las Ciencias Informáticas y en la facultad 3.
- Estudio de los sistemas existentes para la gestión de procesos de cooperación internacional en el país 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.
- Realización de la captura y descripción de los requerimientos asociados a la gestión de trámites de las salidas y visitas hacia y desde el exterior
- Realización de la implementación del sistema.

Validación de los resultados obtenidos.

Posibles resultados:

- Informe de estudio de los sistemas existentes para la gestión de procesos de cooperación internacional en el país a fin de identificar las principales características a tener en cuenta para el desarrollo de la solución.
- Modelos de datos, diagrama de despliegue, diagrama de componentes y artefactos generados por la metodología SXP (Lista de reserva de producto, tarjetas CRC, historias de usuario y las tareas de ingeniería).

- La segunda versión del subsistema de gestión de los procesos de trámites de salidas y visitas, hacia y desde el exterior para la gestión de su información la Facultad 3.

Modelo Metodológico: Métodos de Investigación Científica empleados: Métodos teóricos

El modelo metodológico de este trabajo de diploma contiene los métodos de investigación científica que dieron soporte y contribuyeron al desarrollo de la investigación.

Histórico-lógico: El método histórico estudia la trayectoria real de los fenómenos y acontecimientos en el decursar de su historia. (1)

Este método fue utilizado con el objetivo de estudiar todo lo referente al proceso de cooperación internacionales en la universidad de las ciencias informáticas y en Cuba.

Inductivo-deductivo: La inducción la podemos definir como una forma de razonamiento por medio de la cual se pasa del conocimiento de casos particulares a un conocimiento más general que refleja lo que hay de común en los fenómenos individuales y la deducción es una forma de razonamiento, mediante el cual se pasa de un conocimiento general a otro de menor nivel. (1)

Se hace uso de deducciones para llegar a tener una visión clara de lo que se quiere hacer y adquirir así, nuevos conocimientos.

Analítico - Sintético: Constituyen dos procesos cognitivos. El análisis es una operación intelectual que posibilita descomponer mentalmente un todo complejo en sus partes y cualidades. La síntesis es la operación inversa, que establece mentalmente la unión entre las partes, previamente analizadas y posibilita descubrir relaciones y características generales. (1)

Se hace uso de este método para analizar y sintetizar toda la información correspondiente a la investigación.

Método Modelación: El modelo científico es un instrumento de la investigación de carácter material o teórico, creado por los científicos para reproducir el fenómeno que se está estudiando. El modelo es una reproducción simplificada de la realidad, que cumple una función heurística, ya que permite descubrir y estudiar nuevas relaciones y cualidades del objeto de estudio. (1)

Se hace uso de este método para modelar un proceso real y poder validar la estrategia a proponer.

Métodos empíricos:

Entrevista: La entrevista es una conversación planificada entre el investigador y el entrevistado para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos. (1)

- Este método fue utilizado para dialogar con el asesor de cooperación internacional de la facultad 3, cliente de la investigación. Para lograr un mejor entendimiento del proceso y así conocer su valoración acerca del mismo. Además fue la técnica principal para la captura de requisitos.

Observación: En síntesis podemos entender la observación como el registro visual de lo que ocurre en una situación real, en un fenómeno determinado, clasificando y consignando los acontecimientos pertinentes de acuerdo con algún esquema previsto. (1)

- Este método es utilizado para poseer conocimientos profundos a cerca de los procesos de cooperación internacional.

El trabajo de diploma está estructurado de la siguiente manera:

Capítulo 1: Fundamentación Teórica: Se realiza un estudio de la fundamentación teórica de investigación y de las herramientas, lenguajes y tecnologías a utilizar.

Capítulo 2: Análisis y diseño del subsistema: Se abordan los aspectos funcionales para el desarrollo del sistema, se definen los procesos fundamentales por medio de las historias de usuarios creadas por el cliente. Siguiendo los pasos estipulados por la metodología escogida para el análisis y diseño.

Capítulo 3: Implementación y prueba del subsistema: Este capítulo contiene los aspectos relacionados a la implementación del subsistema. Se exponen las pruebas realizadas para alcanzar la validación satisfactoria de la investigación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

A lo largo del siguiente capítulo se pretende dar una visión global de los fundamentos teóricos, plasmando los conceptos relacionados con la cooperación internacional y el estudio de la información recopilada a lo largo de la investigación, con el fin de lograr un mejor entendimiento de los términos frecuentados en el documento. Por último, se confecciona un análisis de las diferentes tecnologías, lenguajes y herramientas utilizado para el desarrollo de la aplicación.

1.2 Marco conceptual

El concepto de gestión: hace referencia a la acción y a la consecuencia de administrar o gestionar algo. Al respecto, hay que decir que gestionar es llevar a cabo diligencias que hacen posible la realización de una operación comercial o de un anhelo cualquiera. Administrar, por otra parte, abarca las ideas de gobernar, disponer, dirigir, ordenar u organizar una determinada situación. (2)

También y a la par en una gestión habrá que dirigir, gobernar, disponer, organizar y ordenar con el propósito de lograr los objetivos propuestos. De lo dicho se desprende que la gestión es una tarea que requerirá de mucha conciencia, esfuerzo, recursos y buena voluntad para ser llevada a cabo satisfactoriamente.

Una gestión, entonces, podrá estar orientada a resolver un problema específico, a concretar un proyecto, un deseo, pero también puede referir a la dirección y administración que se realiza en una empresa, una organización, un negocio, e incluso a nivel de gobierno, es común que la tarea que lleva a cabo el gobierno de un determinado país sea también denominada como gestión y esta a su vez es uno de los procesos importantes en el área de la cooperación internacional. (2)

Cooperación Internacional: Refiere al conjunto de acciones que derivan de los flujos de intercambio que se producen entre sociedades nacionales diferenciadas en la búsqueda de beneficios compartidos en los ámbitos del desarrollo económico y el bienestar social, o bien, que se desprenden de las actividades que realizan tanto los organismos internacionales que integra el Sistema de las Naciones Unidas como aquellos de carácter regional, intergubernamentales o no gubernamentales, en cumplimiento de intereses internacionales particularmente definidos. La cooperación internacional así descrita se entiende como la

movilización de recursos financieros, humanos, técnicos y tecnológicos para promover el desarrollo internacional. (3)

Los trámites necesarios para efectuar la entrada y la salida desde y hacia el exterior en la UCI poseen un conjunto de pasos para la aprobación de las mismas que incluyen la solicitud por el interesado en recibir al visitante o el interesado en viajar. Esta solicitud puede ser realizada 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 esta.

Otra actividad vinculada al proceso de cooperación internacional es el análisis de la solicitud por parte del área solicitante con los representantes políticos y de masas (Partido Comunista de Cuba, Unión de Jóvenes Comunistas, Federación de Estudiantes Universitarios) y se someta a aprobación de la misión a nivel de facultad en el Consejo de Dirección. Posteriormente se procede al llenado de la planilla de entrada o salida al exterior y ciertos modelos establecidos para este fin. Las planillas anteriormente llenadas se someten a aprobación por la Rectoría: de ser aprobada, la Oficina de Trámites gestiona una serie de documentos imprescindible para la entrada y la salida al exterior de los solicitantes. La duración de esta gestión incluye el tiempo transcurrido desde que se hace la solicitud hasta que esta culmine.

Trámite: Estado de un proceso administrativo por el que tiene que pasar un asunto para ser solucionado, diligencia o gestión. La etimología latina expresa que se trata del paso de una parte a otra. De ahí, cada uno de los estados, pasos y resoluciones o providencias de un asunto.

Proceso: El concepto de proceso se refiere al cambio de estado desde un estado inicial hasta un estado final. Conocer el proceso significa conocer no solo los estados final e inicial sino las interacciones experimentadas por el sistema mientras está en comunicación con su medio o entorno (4).

Si bien es un término que tiende a remitir a escenarios científicos, técnicos y/o sociales planificados o que forman parte de un esquema determinado, también puede tener relación con situaciones que tienen lugar de forma más o menos natural o espontánea.

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; 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 interinstitucionales y contribuir a la identificación y gestión de programas de cooperación internacional. 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 interinstitucionales 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. (5)

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. (6)

Se realizó un estudio de diferentes sistemas vinculados a la cooperación internacional con el objetivo de profundizar en el conocimiento de los mismos para tener mayor dominio a la hora de desarrollar un sistema con la calidad y el funcionamiento requerido por el cliente.

1.3 Sistemas de gestión de procesos de cooperación internacional

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. (7)

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 que en el momento en que fue desarrollado, estas gestiones no se realizaban. (7)

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 puede 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. (7)

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. (7)

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. (7)

En la facultad 3 de la UCI, se realizó un subsistema (SIGECI) para gestionar los procesos de cooperación internacional en la misma, este tenía el fin de integrarse a la Plataforma de Gestión de la Facultad 3 (SO3), pero contaba con el inconveniente que fue realizado en Symfony 1.4 presentando problemas de incompatibilidad con SO3, no se gestionaba el plan de actividades, el módulo de solicitudes se encontraba desactualizado funcionalmente ya que después de un análisis profundo de la primera versión muchas de

funcionalidades que se desarrollan en este módulo no cumplían con las reglas del negocio de acuerdo con lo pedido por el cliente de cooperación internacional, no se manejaba la adecuada gestión de la información para los procesos de trámites de cooperación internacional y por todo lo antes mencionado dificultaba la adecuada gestión de la información de cooperación internacional en la facultad 3.

Por lo antes planteado se decidió realizar una nueva versión de SIGECI para lo cual hubo que realizar un análisis y captura de requisitos a través de la técnica de entrevista con el cliente para darle cumplimiento a las necesidades específicas de cooperación internacional en la facultad 3, para mejorar la gestión de la información y a su vez la nueva versión de SIGECI proporcionara la ayuda necesaria para la cooperación internacional de la facultad 3 ya que este permite la votación en línea lo cual facilitaría el trabajo a los encargados de las tomas de decisiones en la facultad y es un proceso nuevo para la UCI.

1.4 Metodología, Herramientas y tecnologías utilizadas

La decisión de las herramientas a escoger para el desarrollo del presente trabajo estuvo marcada por la línea de trabajo de la plataforma a integrar la segunda versión del subsistema de trámites de entrada y salida desde y hacia el exterior en la facultad 3, para ayudar a que futuros proyectos sean más fáciles de integrar a SO3.

1.4.1 Aplicación web

Las aplicaciones web permiten la generación automática de contenido, la creación de páginas personalizadas según el perfil del usuario o el desarrollo del comercio electrónico. Además, una aplicación web permite interactuar con los sistemas informáticos de gestión de una empresa, como puede ser gestión de clientes, contabilidad o inventario, a través de una página web. (8)

Las aplicaciones web se encuadran dentro de las arquitecturas cliente/servidor: un ordenador solicita servicios (el cliente) y otro está a la espera de recibir solicitudes y las responde (el servidor). En esta investigación se aborda la programación de la parte del cliente de las aplicaciones web. Existen multitud de tecnologías que se pueden emplear para programar las aplicaciones web, como ActiveX o applets, pero no están tan estandarizadas como las dos que se muestran a continuación: HTML y JavaScript. (8)

Ventajas:

- **Ahorra tiempo:** Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.

- **No hay problemas de compatibilidad:** Basta tener un navegador actualizado para poder utilizarlas.
- **No ocupan espacio** en nuestro disco duro.
- **Actualizaciones inmediatas:** Como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
- **Consumo de recursos bajo:** Dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otro ordenador.
- **Multiplataforma:** Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- **Portables:** Es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil...) porque se accede a través de una página web (sólo es necesario disponer de acceso a Internet). La reciente tendencia al acceso a las aplicaciones web a través de teléfonos móviles requiere sin embargo un diseño específico de los ficheros CSS para no dificultar el acceso de estos usuarios.
- **La disponibilidad suele ser alta** porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- **Los virus no dañan** los datos porque éstos están guardados en el servidor de la aplicación.
- **Colaboración:** Gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones online de calendarios u oficina.
- Los navegadores ofrecen cada vez más y mejores funcionalidades para crear aplicaciones web ricas.

1.4.2 Metodología de desarrollo

Una **metodología de desarrollo de software** es un conjunto de pasos y procedimientos que deben seguirse para desarrollar software. Una metodología está compuesta por la división organizada del proyecto en etapas, en escoger de forma efectiva las tareas, aplicar las restricciones específicas, escoger las técnicas y herramientas a emplear, controlar y gestionar el proyecto con efectividad y calidad. El uso de una metodología eficiente puede ofrecer diversas ventajas entre las que podemos citar las siguientes:

(9)

- Existencia de reglas predefinidas
- Cobertura total del ciclo de desarrollo
- Verificaciones intermedias
- Planificación y control
- Comunicación efectiva
- Utilización sobre un abanico amplio de proyectos
- Fácil formación
- Herramientas CASE
- Actividades que mejoren el proceso de desarrollo
- Soporte al mantenimiento
- Soporte de la reutilización de software

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 con el cliente. Según la necesidad que se persigue para el presente trabajo de diploma se escoge la metodología ágil para el desarrollo de la misma por el poco tiempo con el que se dispone para dar la solución final del trabajo del autor. (9)

Una metodología ágil es efectiva para modelar y documentar un proyecto de software, es una colección de valores principios y prácticas para modelar software que puede ser aplicado de manera simple y ligera. Para el desarrollo del trabajo se escoge la metodología ágil SXP ya que la plataforma a integrar la solución final se rige por esta y se pretende mantener una misma línea de trabajo. (9)

SXP

La metodología SXP está compuesta por la metodología XP y SCRUM que es un método adaptativo de gestión de proyectos que se basa en los principios ágiles. SXP ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo. Como principales artefactos que genera el uso de la metodología SXP se encuentran la Lista de reserva de producto, Tarjetas crc, Historias de usuario y las Tareas de ingeniería.

1.4.3 Lenguaje de modelado

UML o lenguaje Unificado de Modelado (por sus siglas en inglés, UnifiedModelingLanguage) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que es un lenguaje, cuenta con reglas para combinar tales elementos. Es importante destacar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema. (10)

1.4.4 Herramientas de Modelado

Se puede definir a las Herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software. Como es sabido, los estados en el Ciclo de Vida de desarrollo de un Software son: Investigación Preliminar, Análisis, Diseño, Implementación e Instalación. (11)

1.4.5 Visual Paradigmfor UML

Como herramienta CASE de modelado con UML se utilizó Visual Paradigmfor UML en su versión 6.4. Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Visual Paradigm es una herramienta creada para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas entre otras opciones. Constituye una herramienta de software libre de gran utilidad para el analista. (12)

1.4.6 Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD

relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos. (13)

PostgreSQL 9.1

Es un sistema de gestión de bases de datos, objeto-relacional (ORDBMS), Está considerado como el sistema de base de datos de código abierto más avanzado del mundo, publicado bajo la licencia BSD. Brinda beneficios de código abierto y estabilidad. El sistema que se desea desarrollar requiere la gestión de altos volúmenes de información, por lo que se necesita un sistema que maneje eficientemente las transacciones y la persistencia de los datos de manera segura. (14)

Se escogió PostgreSQL en su versión 9.1 ya que incluyen características tan avanzadas como la replicación sincrónica para clústeres, permitiendo alta disponibilidad con consistencia a través de múltiples nodos, que asegura que las transacciones han sido confirmadas por una réplica del servidor maestro. También se cuenta con búsquedas más potentes, implementando un “método de indexado de los K vecinos más próximos” las cuales mejoran la velocidad de las búsquedas de forma drástica (indican de sentencias que tardaban 2 segundos han pasado a tardar 20ms). También indican una característica exclusiva de PostgreSQL, que es “hacer cumplir reglas de negocio definidas por el usuario arbitrariamente complejas dentro de la base de datos sin bloqueos, mediante la detección automática de condiciones de carrera en sus transacciones SQL en tiempo de ejecución”, que si bien parece algo poco usable puede jugar un buen rol en el ámbito de la seguridad.

PgAdmin III

PgAdmin III es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL entre ellas: PervasivePostgres, EnterpriseDB, MammothReplicator y SRA PowerGres. Incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad. (15)

1.4.7 IDE de desarrollo

Un entorno de desarrollo integrado o IDE, es un programa informático compuesto por un conjunto de herramientas de programación o puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

NetBeans

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. Con el NetBeans se pueden hacer aplicaciones web o también de escritorio. Entre sus principales características se encuentran mejoras en el editor de código, instalación y actualización más simple, características visuales para el desarrollo web, mejoras para UML y soporte para PHP. (16)

1.4.8 Tecnologías Web.

Las tecnologías web implican un conjunto de herramientas que nos facilitarán lograr mejores resultados a la hora del desarrollo de un sitio web.

JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador aunque existe una forma de JavaScript del lado del servidor. Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, en aplicaciones de escritorio es también significativo. JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Documento del modelo de objetos (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML. (17)

Hojas de estilo en cascada (CSS).

Hojas de Estilo en Cascada es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne. (18)

Lenguaje de Marcas de Hipertexto (HTML).

HTML es el lenguaje con el que se definen las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web. Las etiquetas HTML tienen, por lo general, una etiqueta de apertura y una de cierre; aunque existen algunas excepciones en las que solo basta con colocar la de apertura. (19)

Frameworks

Framework es la aplicación o conjunto de bibliotecas orientadas a la reutilización a muy gran escala de componentes software para el desarrollo rápido de aplicaciones. En esta categoría se incluye la Biblioteca de Componentes Visuales (VCL) de CodeGear, Swing y AWT para el desarrollo de aplicaciones con formularios en Java, Struts, para aplicaciones web también en Java, Ruby OnRails para aplicaciones web con Ruby, y las antiguas Microsoft FoundationClasses y Turbo Vision de Borland. (20)

JQuery 1.8

Es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones (FLV) y agregar interacción con la técnica AJAX a páginas web. JQuery es un software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. JQuery al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. (21)

Características

Selección de elementos DOM.

- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
- Eventos.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.
- AJAX.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes.

Symfony 2.0 framework

Es un framework muy poderoso y con muchísimas experiencias acumuladas. Está desarrollado en PHP5.3, se puede utilizar en plataformas Unix (Unix, Linux) y Windows. Incorpora el patrón MVC, soporta AJAX, plantillas y un gran número de bases de datos. Es un Framework de desarrollo rápido (22).

Symfony 2 soluciona diversos problemas como el trabajo con menos archivos, es mucho más fácil de aprender, más flexible y completo, sustituye los plugins por un nuevo concepto (bundles) y además establece ya como "oficiales" un ORM (Doctrine) y un gestor de versiones (Git).

Mapeado de objetos relacionales. (ORM)

Consiste en una técnica de programación para convertir datos entre el lenguaje de programación orientado a objetos utilizado y el sistema de base de datos relacional utilizado en el desarrollo de aplicaciones. (23)

Propel

Propel es un ORM para PHP que soporta cualquier operación usual en un crud ya sea crear un registro o actualizar los existentes, genera la clase PHP del modelo y lo hace basado en XML, soporta la generación de sus respectivos markups desde una base de datos existente, soporta varios motores de base de datos y también soporta herencia simple.

Doctrine

Doctrine como ya se ha dicho eclipsa a Propel ya que tiene todas las características antes mencionadas de este y además tiene sus propias características como que soporta la aplicación de varios "behaviors" a tus modelos, por ejemplo un modelo Timestampable automáticamente creará 2 columnas: *created_at* y *updated_at*, las cuales guardarán la fecha cuando un registro es creado y cuando un registro es actualizado respectivamente. Doctrine tiene un motor de búsqueda fulltext. Además, Doctrine soporta Data fixtures y migraciones, con lo que podríamos decir que en estas características avanzadas Doctrine supera a Propel.

ORM adoptado

Después de analizar profundamente cada una de las características que ofrece propel y doctrine, se decide optar por doctrine ya que el mismo es el ORM por defecto de Symfony y posee todo el soporte de apoyo de SensioLabs empresa desarrolladora del framework. Por otra parte soporta dos tipos de herencia: simple, donde todas las clases tienen las mismas columnas, y la herencia de agregación, donde almacena un valor adicional en la tabla, permitiendo la instanciación automática del tipo de modelo correcto cuando se realiza una consulta.

HipertextPreprocesor (PHP)

PHP es un lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo (24).

Para el desarrollo de la aplicación se escoge utilizar PHP en su versión 5.3 ya que es la compatible con el framework Symfony 2.0 y presenta características como la mejora de rendimiento, corrección de errores y la aproximación a la programación orientada a objetos.

Servidor Web

La principal función de un servidor Web es almacenar los archivos de un sitio y emitirlos por Internet para poder ser visitado por los usuarios. Básicamente, un servidor Web es una gran computadora que guarda y transmite datos vía Internet. Cuando un usuario entra en una página de Internet su navegador se comunica con el servidor enviando y recibiendo datos que determinan qué es lo que ve en la pantalla. Por eso decimos que los servidores Web están para almacenar y transmitir datos de un sitio según lo que pida el navegador de un visitante. (25)

Apache

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Entre sus características destacan (26):

- Multiplataforma
- Es un servidor de web conforme al protocolo HTTP/1.1
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Basado en hebras en la versión 2.0

- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

1.5 Conclusiones

A partir del estudio realizado como parte del marco teórico se dio a conocer un conjunto de elementos que marcan el punto de partida sobre el cual se apoyará la investigación para iniciar la construcción de un software que cumpla con los objetivos propuestos. Se realizó un estudio intenso sobre los diferentes sistemas de cooperación internacional existentes tanto a nivel de la Universidad de las Ciencias Informáticas como a nivel nacional llegando a la decisión de desarrollar un software que cumpla con las necesidades actuales principalmente de la facultad 3. Se han expuesto las herramientas y metodologías necesarias para el análisis y desarrollo de los demás capítulos. Se han seleccionado y justificado las herramientas a tener en cuenta en el desarrollo del software propuesto.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SUBSISTEMA

2.1 Introducción

El presente capítulo tiene como objetivo el análisis y diseño de la solución informática del Subsistema para la gestión de los procesos de trámites de salidas y visitas, hacia y desde el exterior, en La Facultad 3. A partir del análisis del negocio se obtiene la lista de reserva del producto la cual cuenta con los requisitos a informatizar, así como la descripción de los mismos. Acatando la metodología ágil SXP se generan los artefactos pertinentes.

2.2 Lista de reserva del producto

Es una lista priorizada que define el trabajo que se va a realizar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto. Esta lista puede crecer y modificarse a medida que se obtiene más conocimiento acerca del producto y del cliente. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto. (27)

Tabla 1. Lista de reserva del producto.

No	Requisito	Descripción	Complejidad
1	Gestionar solicitud interna	Se gestiona la solicitud por parte de un individuo perteneciente a la UCI, se modifica, elimina o listan las solicitudes y sus estados.	Alta
2	Gestionar solicitud externa	Se gestiona la solicitud por parte de un individuo exterior a la UCI, se modifica, elimina o listan las solicitudes y sus estados.	Alta
3	Mostrar Solicitud	Se muestran todos los detalles de la solicitud creada por el solicitante.	
4	Revisar	La solicitud creada pasa por la revisión de las	Alta

	solicitud	contra-partes (jefe de área, investigación, postgrado y cooperación), se proveen criterios a tener en cuenta para los votantes.	
5	Gestionar prórroga	Requisito que prolonga la estancia de la misión, argumentando los motivos de esta.	Alta
6	Gestionar informe de viaje	Requisito que adiciona, modifica o elimina un informe de viaje.	Alta
7	Gestionar costo	Requisito encargado de llevar el inventario de la solicitud.	Alta
8	Realizar votación	Requisito que realiza la votación sobre cada solicitud existente, tiene incluida la realización del voto especial sobre una solicitud.	Alta
9	Cerrar votación	Se cierran las votaciones de las solicitudes que llevan más del tiempo previsto para la realización de las mismas.	Alta
10	Ver estado de votación	Muestra el estado en el cual se encuentra la votación.	Alta
11	Gestionar visita	Se gestiona la solicitud por parte de un individuo de otro país que desea realizar una visita a la UCI.	Media
12	Mostrar visita	Se muestran los datos de la persona que realizará la visita a la universidad	Media
13	Gestionar	Se muestran y escogen los servicios a los que tiene	Media

	servicio	derecho el visitante.	
14	Gestionar regalos	Se le asigna el regalo al visitante.	Media
15	Gestionar entes	Se añaden los entes a participar, asignándole la responsabilidad que desempeñará en el proceso de votación.	Media
16	Gestionar persona externa	Se manejan los datos de la persona externa a la que se le hará la solicitud de misión.	Media
17	Gestionar institución	Se maneja la institución que representa el visitante al igual que el tipo de institución.	Media
18	Gestionar estado de solicitud	Se maneja por parte del asesor de cooperación internacional el estado de la solicitud.	Baja
19	Gestionar tipo de misión	Se maneja por parte del asesor de cooperación internacional el tipo de misión.	Baja
20	Gestionar regla de votación	Se maneja por parte del asesor de cooperación internacional la regla de votación.	Baja
21	Gestionar tipo de institución	Se maneja por parte del asesor de cooperación internacional el tipo de institución.	Baja

22	Gestionar rol	Se maneja por parte del asesor de cooperación internacional el rol que desempeñará el solicitado en la misión.	Baja
23	Gestionar estado de voto	Se maneja por parte del asesor de cooperación internacional los diferentes estados por los que pasara el proceso de votación.	Baja
24	Gestionar concepto de costo	Se maneja por parte del asesor de cooperación internacional el concepto de costo.	Baja
25	Gestionar implicado en costo	Se maneja por parte del asesor de cooperación internacional el implicado en costo.	Baja

Requisitos no funcionales.

Los requerimientos 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. (28)

Tabla 2.Requisitos no funcionales.

No	Tipo	Descripción
1	Autenticación	Los usuarios se autenticarán haciendo uso del servicio LDAP que consume la plataforma SO3 y brinda a SIGECI, la administración de los roles será por la aplicación correspondiente a la plataforma SO3.
2	Seguridad	Confiable: La información manejada por el sistema estará

		protegida de accesos no autorizados y divulgación.
3	Seguridad	Integridad: La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos.
4	Seguridad	Disponibilidad: Solo tendrán acceso al sistema el jefe cooperación internacional, el asesor principal o la persona facultada para hacer uso de sus funcionalidades.
5	Licencia	El sistema utiliza la política de software libre, todas las herramientas que se utilizan son software libre. Para la herramienta Visual Paradigm se utiliza la licencia que la Universidad posee.
6	Soporte	Realizar las pruebas, e instalaciones necesarias para lograr el mejoramiento del subsistema.
7	Usabilidad	El software tendrá siempre la posibilidad de ayuda disponible para cualquier tipo de usuario, lo que le permitirá un avance considerable en la explotación de la aplicación en todas sus funcionalidades. Tiene que poseer una interfaz agradable para el cliente.
8	Apariencia o interfaz externa	El sistema tiene que ofrecer una interfaz amigable, fácil de operar. El sistema tiene que mantener la línea de diseño establecida para la institución que mantiene la uniformidad y

		representatividad de la misma. Las interfaces tienen que mostrar un diseño sencillo, con pocas entradas, permitiendo un balance adecuado entre funcionalidad y simplicidad de tal manera que no se haga difícil para los usuarios la utilización del sistema.
9	Requerimientos de software	Linux, Windows XP o superior.
10	Requerimientos de hardware	Microprocesador AMD, Memoria RAM de 4GB, 5GB de espacio en disco o más.

2.3 Historia de usuarios (HU)

Las Historias de usuarios (HU) son tarjetas desarrolladas desde la perspectiva del cliente, es el artefacto que sustituye a los casos de uso de la metodología tradicional, su objetivo es especificar en detalle las necesidades del sistema, son descripciones generalmente cortas y sin terminología técnica (27). Se elaboraron 25 HU cada una de ellas respondiendo a las diferentes funcionalidades deseadas por el cliente. A continuación se expone la HU Gestionar solicitud interna. Las demás se pueden adquirir en el expediente de proyecto.

Tabla 3. Historia de usuario Gestionar solicitud interna

Historia de Usuario	
Número:HU_1	Nombre Historia de Usuario: Gestionar solicitud interna
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Solicitante	Iteración Asignada:1

Prioridad en Negocio: Alta	Puntos Estimados: 5 días
Riesgo en Desarrollo: Alta	Puntos Reales: 5 días
<p>Descripción:</p> <p>Los usuarios se autentican en el sistema. El sistema valida el permiso de administración del usuario. Muestra la interfaz de solicitud, en la cual se encuentra todos los datos referentes a las solicitudes, la misma se puede eliminar, modificar y adicionar una solicitud interna.</p> <p>Para adicionar una solicitud interna se necesita introducir los siguientes datos:</p> <ul style="list-style-type: none"> - Buscar una persona interna a partir del carnet de identidad, mediante un servicio que brinda la plataforma SO3. -Foto: foto del solicitado (se carga automáticamente). <p>El usuario Introduce los datos restantes:</p> <ul style="list-style-type: none"> - Nombre de misión: nombre de la misión solicitada. - Correo electrónico: dirección de correo electrónico del solicitado (se introduce manualmente). - Solicitado por: persona encargada de realizar la solicitud. -País destino: país al cual el solicitado deberá viajar (se selecciona). -Fecha de salida: fecha de salida del solicitado (se selecciona). -Fecha de regreso: fecha de regreso del solicitado (se selecciona). -Importancia del viaje: motivo por el cual debe viajar el solicitado (se introduce). -Tipo de misión: tipo de misión que debe cumplir el solicitado (se selecciona). -Organizaciones consultadas: son las organizaciones políticas a la que pertenece(se 	

marca si fueron consultadas).

-Responsabilidades asumidas: se marca si las responsabilidades que desempeña fueron asumidas.

-Sede principal: es la sede de la ciudad a la que viajará.

-Acuerdos CD: número del acuerdo en el acta del consejo de dirección.

-Objetivos: objetivo de la misión que va a desempeñar.

-Fundamentación: fundamentación de la misión.

-Observaciones: datos adicionales que se quieran introducir.

Se registra la solicitud, se confirma el registro de los datos a través de un mensaje "Proceso realizado satisfactoriamente" y se agregan los datos del solicitado en la interfaz Gestionar Solicitud.

Observaciones:

2.4 Plan de Iteraciones.

Las HU seleccionadas para cada iteración son desarrolladas y probadas en un ciclo de iteración, de acuerdo con el orden preestablecido. Cada HU se traduce en tareas específicas de programación. Este plan de iteraciones ayuda al avance y organización de la implementación. El mismo se dividirá en 3 iteraciones en dependencia del riesgo en desarrollo de las HU.

Iteración 1: En la primera iteración se implementarán las HU con prioridad baja, obteniendo al final de la misma una primera versión de prueba, dando al sistema las primeras funcionalidades.

Iteración 2: En la segunda iteración se implementarán las HU con prioridad de negocio media, donde se encuentran la mayoría de las funcionalidades del sistema como son la gestión de contactos, revisión de solicitudes entre otras.

Iteración 3: En la tercera iteración después de haberse implementado las funcionalidades específicas y principales se elaboran las últimas HU con prioridad alta relacionadas con los procesos de solicitudes y votaciones.

Plan de duración de las iteraciones: Una vez definidas las HU a implementar en cada una de las iteraciones, se procede a definir el tiempo de duración de cada una de las historias creando el plan de duración de iteraciones que propone la metodología utilizada. Este plan tiene como finalidad mostrar la duración de cada iteración y el orden en que las historias de usuarios serán implementadas.

Tabla 4. Plan de iteraciones

Iteración	Orden de las HU a implementar	Duración total de las iteraciones
Iteración 1	<ol style="list-style-type: none"> 1. Gestionar estado de solicitud 2. Gestionar tipo de misión 3. Gestionar regla de votación 4. Gestionar tipo de institución 5. Gestionar rol 6. Gestionar estado de voto 7. Gestionar concepto de costo 8. Gestionar implicado en costo 	3 semanas
Iteración 2	<ol style="list-style-type: none"> 1. Gestionar visita 2. Mostrar visita 3. Gestionar servicio 4. Gestionar regalos 5. Gestionar entes 6. Gestionar persona externa 7. Gestionar institución 	4 semanas

Iteración 3	1	Gestionar solicitud interna	7 semanas
	2	Gestionar solicitud externa	
	3	Mostrar solicitud	
	4	Revisar solicitud	
	5	Gestionar prórroga	
	6	Gestionar informe de viaje	
	7	Gestionar costo	
	8	Realizar votación	
	9.	Ver estado de votación	

2.5 Patrones usados en el diseño

El diseño fue elaborado siguiendo patrones basados en la experiencia que de manera general constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos.

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC (Modelo, Vista, Controlador), que está formado por tres niveles:

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

El principio más importante de la arquitectura MVC es la separación del código del programa en tres capas, dependiendo de su naturaleza. La lógica relacionada con los datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador. La programación se puede simplificar si se utilizan otros patrones de diseño, de esta forma, las capas del modelo, la vista y el controlador se pueden subdividir en más capas. (24)

Para la implementación de Symfony se utilizan varios patrones, situándolos en las capas de Modelo y Control que plantea el patrón arquitectónico MVC.

Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos. Los patrones GRASP que se utilizaron son los siguientes:

Experto: Dicho patrón es evidenciado en la definición de las clases de acuerdo a las funcionalidades que deben realizar a partir de la información manejada dentro del componente, como por ejemplo las clases controladoras y las del modelo. La cual gestiona la configuración de las solicitudes, será la responsable de efectuar las operaciones que conciernen a las funciones: adicionar, modificar, eliminar y actualizar las solicitudes, asumiendo toda la lógica para cada una de ellas. Este patrón se evidencia en la clase `PersonaExternaController` ya que tiene los atributos y los métodos que le corresponden.

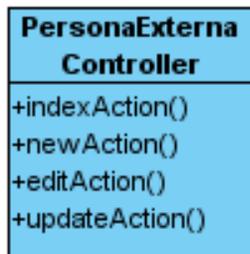


Ilustración 1. Clase `PersonaExternaController`

Creador: Se utiliza para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la información necesaria para ello. Este patrón se evidencia en la clase `PersonaExternaGestor` que es la encargada de crear todas las instancias de la entidad persona.

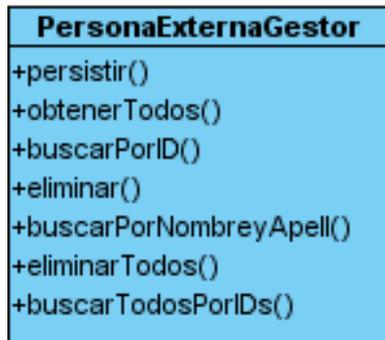


Ilustración 2. Clase PersonaExternaGestor

Alta cohesión: Se utiliza para brindar una alta cohesión funcional que evite contener clases con un alto grado de abstracción. Las clases fueron diseñadas de forma tal que contengan las responsabilidades necesarias colaborando con otras para llevar a cabo una tarea. En el componente se tienen las clases controladoras que se encargan de ejecutar acciones de acuerdo a las peticiones que le llegan desde la vista y las clases de acceso a datos que interactúan con el modelo, de forma tal que se elimina la sobrecarga de funcionalidades en las clases controladoras. Este patrón permitirá que las clases sean fáciles de mantener, entender y reutilizar. Se evidencia entre la clase PersonaExternaController, PersonaExternaGestor y persona, donde la clase controller contiene una instancia de la clase persona que se le pasa como parámetro a un método de la clase PersonaExternaGestor.

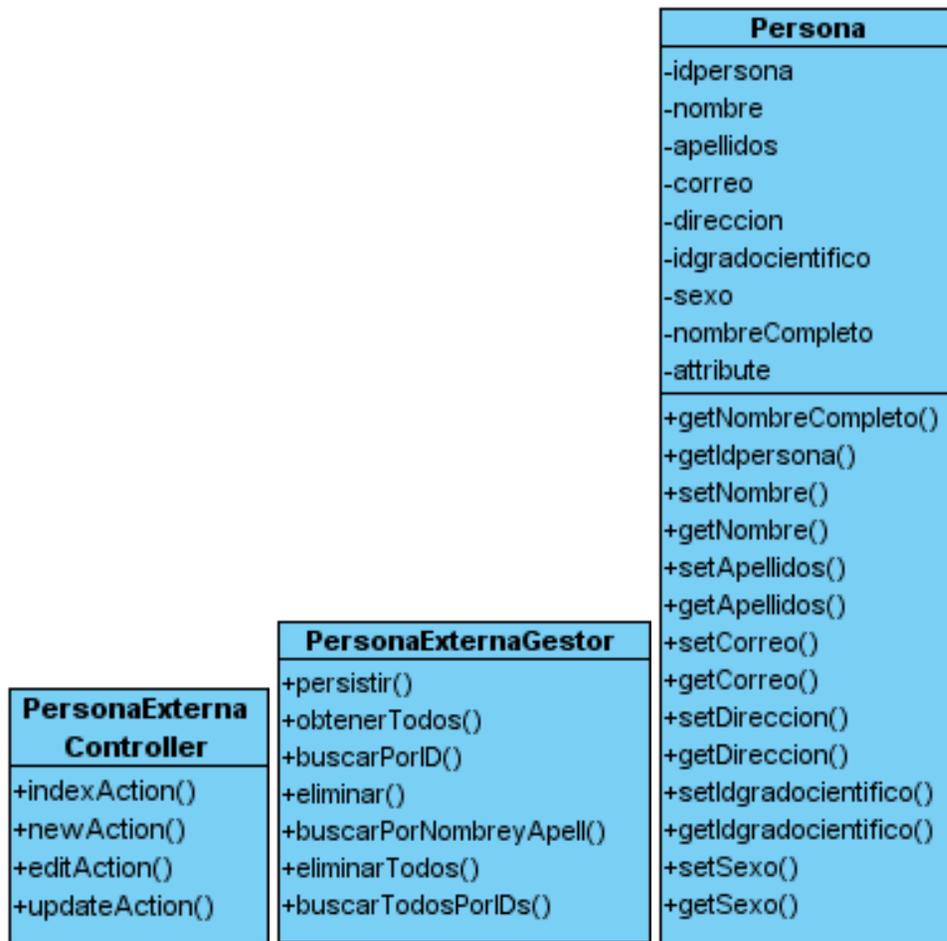


Ilustración 3. Clases PersonaExternaController, PersonaExternaGestor y Persona.

Controlador: Se aplica para asignar la responsabilidad del manejo de los eventos del sistema y definir sus operaciones, es el intermediario entre una determinada interfaz y el algoritmo que la implementa. En todas las clases controller ya que en symfony 2.0 todos los eventos que ocurren tienen que pasar por estas clases.

Bajo Acoplamiento: Se utiliza para estimular la asignación de responsabilidades de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento, de esta forma una clase con bajo acoplamiento tendrá un número mínimo de dependencia con otras clases. En el desarrollo de componente las clases controladoras sólo dependerán de un único controlador frontal para realizar sus funcionalidades y las clases de acceso a datos solo se acoplarán a la

clase modelo que la usa, así se pueden realizar cambios en cada clase de forma independiente. El uso de los patrones Experto y Creador contribuyen al bajo acoplamiento entre las clases del sistema.

Patrones GOF.

Ya que se usan varios frameworks de desarrollo y estos a su vez implementan independientemente sus propios patrones de diseños, a continuación se presenta una breve descripción de los principales patrones empleados.

Singleton (Instancia única): Permite la existencia de una única instancia para una clase permitiendo un mayor rendimiento de la misma y consistencia del código fuente. Se evidencia cuando se accede al `$this->getEntityManager()` que maneja doctrine 2.0.

```
public function buscarTodosPorIDs($ids) {  
  
    $em = $this->getEntityManager();  
  
    $sql = "SELECT i FROM SigeciBundle:Concepto i  
          WHERE i.idconcepto IN(:ids)";  
  
    $dql = $em->createQuery($sql);  
  
    $dql->setParameter('ids', $ids);  
  
    return $dql->getResult();  
}
```

Ilustración 4. Método buscarTodosPorIDs.

Controlador Frontal: El patrón Controlador Frontal podría dividir la funcionalidad en 2 diferentes objetos: el Controlador Frontal y el Dispatcher. En ese caso, El Controlador Frontal acepta todos los requerimientos de un cliente y realiza la autenticación, y el Dispatcher direcciona los requerimientos a manejadores apropiados. Se evidencia en la clase SigeciController de la cual heredan todos los controladores de la aplicación.

SigeciController
+indexAction() +nomAction()

Ilustración 5. Clase SigeciController.

2.6 Tarjetas crc

Las Tarjetas crc (clase, responsabilidad y colaboración) son técnicas de modelado creadas para ayudar a los desarrolladores de software a crear diseños de clases orientados a responsabilidades. Dichas tarjetas constan de tres secciones, nombre de la clase, responsabilidades y colaboradores. La sección responsabilidades se listan cada una de las funciones o tareas que debe ser capaz cumplir un objeto de dicha clase mientras que la sección colaboradora contiene otras clases del diseño. (27)

A continuación se muestra la Tarjeta crc del módulo 1 Nomencladores en el expediente de proyecto se puede apreciar el otro módulo.

Tabla 5. Tarjeta crc módulo 1 Nomencladores

Módulo 1 Nomencladores		
Clases	Descripción	Clases relacionadas
ImplicadoController	Contiene las funcionalidades para eliminar, actualizar y crear un implicado en costo. Posee los métodos indexAction, newAction, createAjaxAction,editAction, updateAjaxActionydeleteAction.	Implicado
EstadoSolicitudController	Contiene las funcionalidades para eliminar, actualizar y crear un estado de solicitud. Posee los métodos indexAction, newAction,	EstadoSolicitud

	createAjaxAction,editAction, updateAjaxActionydeleteAction.	
ReglaVotacionController	Contiene las funcionalidades para eliminar, actualizar y crear una regla de votación. Posee los métodos indexAction, newAction, createAjaxAction,editAction, updateAjaxAction, deleteAction y activarAction.	ReglaVotacion
TipoInstitucionController	Contiene las funcionalidades para eliminar, actualizar y crear un tipo de institución. Posee los métodos indexAction, newAction, createAjaxAction,editAction, updateAjaxActionydeleteAction.	TipoInstitucion
TipoMisionController	Contiene las funcionalidades para eliminar, actualizar y crear un tipo de misión. Posee los métodos indexAction, newAction, createAjaxAction,editAction, updateAjaxActionydeleteAction.	TipoMision
ConceptoController	Contiene las funcionalidades para eliminar, actualizar y crear un concepto de costo. Posee los métodos indexAction, newAction, createAjaxAction,editAction, updateAjaxActionydeleteAction.	Concepto
RegaloController	Contiene las funcionalidades para eliminar, actualizar y crear un regalo.	Regalo

	Posee los métodos indexAction, newAction, createAjaxAction,editAction, updateAjaxActionydeleteAction.	
RolController	Contiene las funcionalidades para eliminar, actualizar y crear un rol. Posee los métodos indexAction, newAction, createAjaxAction,editAction, updateAjaxActionydeleteAction.	Rol
EstadoVotoController	Contiene las funcionalidades para eliminar, actualizar y crear el estado de un voto. Posee los métodos indexAction, newAction, createAjaxAction,editAction, updateAjaxActionydeleteAction.	EstadoVoto

2.7 Modelo de datos.

En el proceso de abstracción que conduce a la creación de una base de datos, desempeña una función prioritaria el modelo de datos. El modelo de datos, como abstracción del universo de discurso, es el enfoque utilizado para la representación de las entidades y sus características dentro de la base de datos. Es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia. (29)

El modelo de datos propuesto en la solución cuenta con un total de 31 tablas. Para su construcción se tuvo en cuenta la reducción a la mínima expresión de los campos nulos y la persistencia de campos resúmenes para agilizar recuperaciones frecuentes de algunos datos que son complejos de calcular. Como tablas principales se tiene la tabla solicitud que tendrá como tablas hijas solicitud interna y solicitud externa.

2.8 Conclusiones

Con el apoyo de la metodología se establece la base para la implementación de la solución. Logrando alcanzar una estructura de la dinámica de organización de los procesos identificados del negocio con el que se alcanzan cada una de las fases de gestión de cooperación. También se obtuvo el análisis y diseño de la solución informática para el mejor funcionamiento de la gestión de los procesos de trámites de salidas y visitas, hacia y desde el exterior, en La Facultad 3.

CAPÍTULO 3: IMPLMENTACIÓN Y PRUEBA

3.1 Introducción

En este capítulo se muestra la implementación que sitúa en práctica el diseño de la solución realizado en el capítulo anterior y se especifica el conjunto de validaciones y pruebas que evalúan la calidad del sistema.

3.2 Tareas de ingeniería

Las tareas de ingeniería son un conjunto de acciones que se realizan con el objetivo de resolver las historias de usuarios, una historia de usuario puede tener una o más tareas de ingeniería en dependencia de la complejidad de la funcionalidad a desarrollar. (27)

En las siguientes tablas se muestran las tareas de ingenierías correspondientes a la HU Gestionar Solicitud Interna. Las restantes tareas de ingenierías pueden ser valoradas en el expediente de proyecto.

Tabla 6. Tarea de Ingeniería 1 de la Historia de Usuario Gestionar Solicitud Interna

Tarea de Ingeniería	
Número Tarea: <i>1</i>	Número Historia de Usuario: <i>1</i>
Nombre Tarea: Diseñar interfaz para gestionar solicitudes internas	
Tipo de Tarea: Diseño	Puntos Estimados: <i>1 día</i>
Fecha Inicio: 17/5/2013	Fecha Fin: 17/5/2013
Programador Responsable : Alfredo Álvarez Nápoles	
Descripción: <i>Diseñar una interfaz capaz de realizar todo el proceso para la gestión de las solicitudes internas.</i>	

Tabla 7. Tarea de Ingeniería 2 de la Historia de Usuario Gestionar Solicitud Interna

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: 1
Nombre Tarea: Desarrollar la interfaz para gestionar solicitudes internas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3 días
Fecha Inicio: 18/5/2013	Fecha Fin: 20/5/2013
Programador Responsable : Alfredo Álvarez Nápoles	
Descripción: <i>Desarrollar una interfaz que sea capaz de adicionar, modificar y eliminar una solicitud interna.</i>	

Tabla 8. Tarea de Ingeniería 3 de la Historia de Usuario Gestionar Solicitud Interna

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: 1
Nombre Tarea: Validar campos de la interfaz para gestionar solicitudes internas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1 día
Fecha Inicio: 21/5/2013	Fecha Fin: 21/5/2013
Programador Responsable : Alfredo Álvarez Nápoles	
Descripción: <i>Verificar que se introduzcan los datos correspondientes a la solicitud interna.</i>	

3.3 Diagrama de componentes

Los diagramas de componentes son otra forma de representar una vista estática del sistema, que representa la organización y dependencia entre los componentes físicos que se necesitan para ejecutar la aplicación, sean estos componentes de código fuente, librerías, binarios o ejecutables. (28)

El diagrama de componentes presentado para la solución cuenta con el componente SO3 que brinda servicios al componente SIGECI y este los consume, dichos servicios serían getPaís, getPaíses, getPersona y getGradoCientifico.

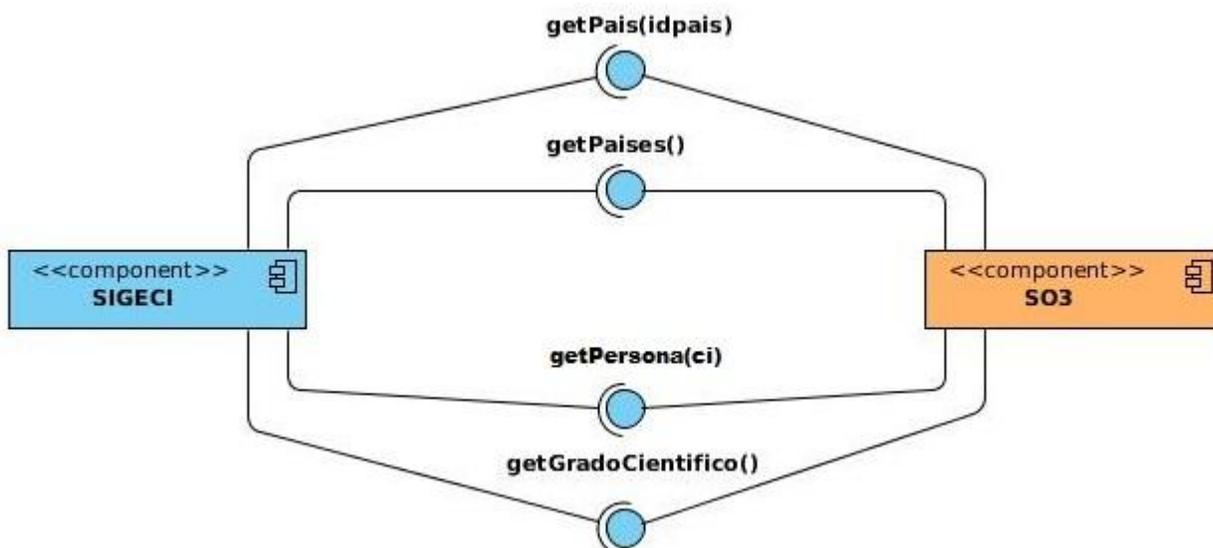


Ilustración 7. Diagrama de componente

3.4 Diagrama de Despliegue.

Los diagramas de despliegue muestran nodos, conexiones, componentes y objetos. Los nodos representan objetos físicos con recursos computacionales como procesadores y periféricos; pueden mostrarse como una clase (una familia de procesadores) o una instancia, por lo que su nombre sigue la misma sintaxis establecida para clases y objetos. Las conexiones son asociaciones de comunicación entre los nodos, y se etiquetan con un estereotipo que identifica el protocolo de comunicación o la red utilizada. Los componentes son archivos de código ejecutable, que residen y se ejecutan dentro de un nodo; se

pueden representar relaciones de dependencia entre los componentes que, de manera similar a las dependencias entre paquetes, corresponden al uso de servicios. (19)

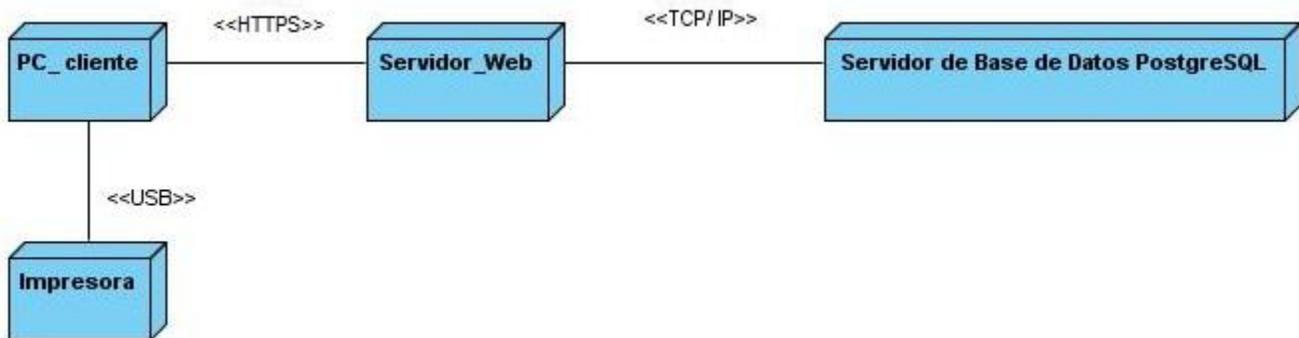


Ilustración 8. Diagrama de Despliegue.

Se tiene el uso de una PC cliente que tendrá conexión con un servidor web a través de la conexión HTTPS, a su vez estará conectada a un servidor de base de datos a través de la TCP/IP y teniendo una impresora conectada a través de un puerto USB a la PC cliente.

3.5 Pruebas

Uno de los pilares fundamentales de SXP es el proceso de pruebas, el cual anima a los desarrolladores a probar constantemente y tanto como sea posible. Mediante esta filosofía se reduce el número de errores no detectados, así como el tiempo entre la introducción de éste en el sistema y su detección, esto contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones. Esta metodología divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente. (28)

3.5.1 Principios de la prueba

Antes de la aplicación de métodos para el diseño de casos de prueba efectivos, se deberá entender ciertos principios básicos que guían las pruebas del software.

- La prueba puede ser usada para mostrar la presencia de errores, pero nunca de su ausencia.
- La principal dificultad del proceso de prueba es decidir cuándo parar.
- Evitar casos de pruebas no planificados, no reusables y triviales a menos que el programa sea verdaderamente sencillo.
- Una parte necesaria de un caso de prueba es la definición del resultado esperado.
- Los casos de pruebas tienen que ser escritos no solo para condiciones de entrada válidas y esperadas sino también para condiciones no válidas e inesperadas.
- Los casos de pruebas tienen que ser escritos para generar las condiciones de salida deseadas.
- El número de errores sin descubrir es directamente proporcional al número de errores descubiertos.
- Las pruebas deberían empezar por lo pequeño y progresar hacia lo grande.
- Con la excepción de las pruebas de unidad e integración, un programa deberá ser experimentado por la persona u organización que lo desarrolló.
- Asigna el programador más creativo a la prueba.

3.5.2 Niveles de Prueba

La Prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. Se distinguen los siguientes niveles de pruebas: (30)

- Prueba de desarrollador
- Prueba independiente
- Prueba de Unidad
- Prueba de Integración
- Prueba de sistema
- Prueba de aceptación

Prueba de unidad: Es la prueba enfocada a los elementos testeables más pequeños del software. Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos, y que ellos funcionen como se espera. La prueba de unidad siempre está orientada a caja blanca. Antes de iniciar cualquier otra prueba es preciso probar el flujo de datos de la interfaz del componente. Si los datos no entran correctamente, todas las demás pruebas no tienen sentido. El diseño de casos de prueba de una unidad comienza una vez que se ha desarrollado, revisado y verificado en su sintaxis el código a nivel fuente.

Prueba de aceptación: Prueba de aceptación del usuario es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido. Un comentario sobre la secuencia de los niveles de prueba. Las pruebas de unidad son implementadas en la iteración más temprana como el primer nivel de prueba. Pero en un proceso iterativo ejecutar todas las pruebas de unidad antes de pasar a niveles siguientes de prueba como regla es inapropiada. Una mejor estrategia es identificar las pruebas de unidad, integración y sistema que ofrecen mayor potencial para encontrar errores y entonces implementarlas y ejecutarlas.

3.5.3 Tipos de prueba de Caja Blanca y Negra

Cualquier producto de ingeniería puede ser probado de una de estas formas: (28)

1. Conociendo la funcionalidad específica para la cual fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa.
2. Conociendo el funcionamiento del producto se pueden desarrollar pruebas que aseguren que “todas las piezas encajen”, o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada.

El 1er enfoque se denomina Prueba de Caja Negra y el 2do Prueba de Caja Blanca.

Prueba de caja negra: Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software.

Prueba de caja blanca: Se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que examinen que están correctas todas las condiciones y/o bucles para determinar si el estado real coincide con el esperado o afirmado. Esto genera gran cantidad de caminos posibles por lo que hay que dedicar esfuerzos a la determinación de las condiciones de prueba que se van a verificar.

3.6 Validación de la implementación.

Partiendo del estudio de la metodología aplicada se seleccionan las pruebas unitarias y de aceptación a fin de obtener la validación del subsistema para la gestión de los procesos de trámites de salidas y visitas, hacia y desde el exterior, en la Facultad 3 versión 2.0.

3.6.1 Evaluación de la ejecución de las pruebas Unitarias.

A continuación se describe el caso de prueba unitaria aplicando caja blanca para el método `costoscreateAction`.

Prueba del camino básico.

La prueba del camino básico es una técnica de prueba de la Caja Blanca propuesta por Tom McCabe. Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el Grafo de Flujo asociado y se calcula su complejidad ciclomática. (28)

Los pasos que se siguen para aplicar esta técnica son:

- A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
- Se calcula la complejidad ciclomática del grafo.
- Se determina un conjunto básico de caminos independientes.
- Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Los casos de prueba derivados del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

Notación de Grafo de Flujo.

Para aplicar la técnica del camino básico se debe introducir una sencilla notación para la representación del flujo de control, el cual puede representarse por un Grafo de Flujo. Cada nodo del grafo corresponde a una o más sentencias de código fuente. Todo segmento de código de cualquier programa se puede traducir a un Grafo de Flujo. A continuación se muestra la sentencia de código del método `costoscreateAction` enumerada y posteriormente el Grafo de Flujo correspondiente a este método. (28)

```

public function costoscreateAction() {
    $request = $this->getRequest(); 1
    $entity = new Costo(); 1
    $form = $this->createForm(new CostoType(), $entity); 1
    $form->bindRequest($request); 1
    $negocio = $this->obteberNegocio(); 1
    if ($this->get("session")->get("idsolicitud")) { 2
        $idsolicitud = $this->get("session")->get("idsolicitud"); 3
        $costos = $negocio->buscarCostosByIDSolicitud($idsolicitud); 3
        $concepto = $entity->getIdconcepto(); 3
        $implicado = $entity->getIdimplicado(); 3
        if ($this->tieneCosto($costos, $concepto, $implicado)) { 4
            $this->get("session")->setFlash("info", 'Ya existe un costo por este concepto e implicado.
            Debe eliminar el anterior para poder insertar uno nuevo. '); 5
            return $this->redirect($this->generateUrl("solicitud_costosnew")); 5
        }
    } else { 6
        $this->get("session")->setFlash("info", 'Por favor, seleccione un elemento de la lista. '); 7
        return $this->redirect($this->generateUrl("solicitud_index")); 7
    }
    if ($form->isValid()) { 8
        $negocio->persistirCosto($entity); 9
        $this->get("session")->setFlash("success", 'La operaci3n ha sido realizada satisfactoriamente. '); 9
        return $this->redirect($this->generateUrl("solicitud_costosnew")); 9
    } else { 10
        $this->get("session")->setFlash("info", 'Ha ocurrido un error al realizar esta operaci3n. '); 11
        return $this->redirect($this->generateUrl("solicitud_index")); 11
    }
}
} 12
}

```

Ilustraci3n 9. Prueba de caja blanca del m3todo costoscreateAction

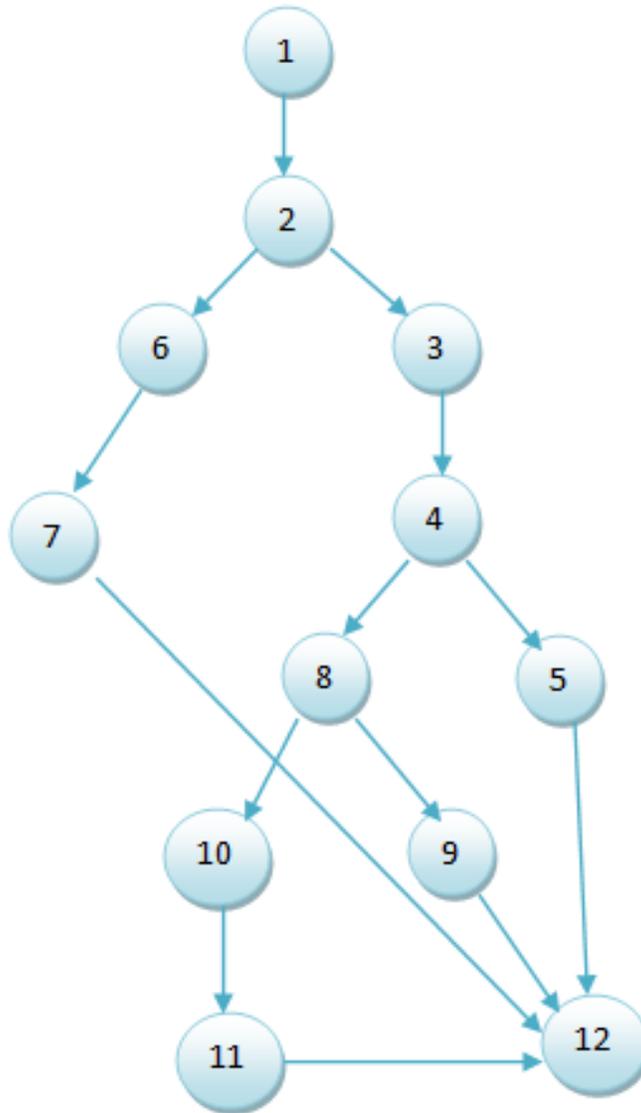


Ilustración 10. Grafo de Flujo Complejidad Ciclomática

La complejidad ciclomática es una métrica de software extremadamente útil pues proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y nos da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez. (28)

El número de regiones del grafo de flujo coincide con la complejidad ciclomática. La complejidad ciclomática, $V(G)$, se define como:

$$V(G) = A - N + 2$$

Donde: A es el número de aristas del grafo y N es el número de nodos.

$$V(g) = A - N + 2$$

$$V(g) = 14 - 12 + 2$$

$$V(g) = 4$$

Camino Independiente

Un camino independiente es cualquier camino del programa que introduce por lo menos un nuevo conjunto de sentencias de procesamiento o una nueva condición. El camino independiente se debe mover por lo menos por una arista que no haya sido recorrida anteriormente. Como resultado final del análisis del Grafo de Flujo generado por el método `costoscreateAction` se obtiene los siguientes caminos independientes, los cuales son recorridos para cualquier x arrojando siempre resultados positivos (28):

Camino 1: 1,2,6,7,12

Camino 2: 1,2,3,4,5,12

Camino 3: 1,2,3,4,8,9,12

Camino 4: 1,2,3,4,8,10,11,12

3.6.2 Evaluación de la ejecución de las pruebas de aceptación.

Estas pruebas se crean a partir de las historias de usuario. Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de estas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable. Se efectuaron pruebas de aceptación según el plan de iteración en tres ocasiones arrojando resultados de un 56% de aceptación en la primera iteración ya que el cliente detectó 11 no conformidades (NC), en la segunda de un 76% encontrándose 6 NC y en la última a un 100% obteniendo un máximo de aceptación por parte del cliente después de realizar las pruebas pertinentes. En gráfica se establece una comparación de los resultados obtenidos por iteraciones.

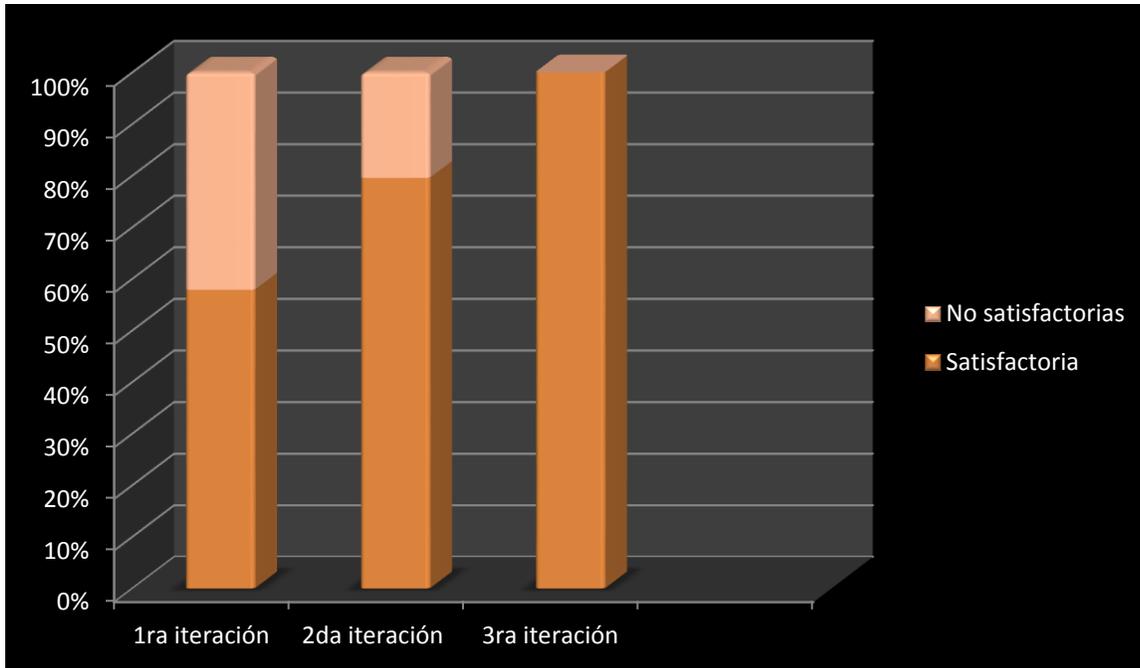


Tabla 9. Gráfica de resultados por iteraciones

A continuación se muestra una de las pruebas de aceptación efectuadas todas con resultados satisfactorios las demás se pueden apreciar en el expediente de proyecto.

Caso de Prueba de Aceptación	
Código Caso de Prueba: <i>SIGECI88</i>	Nombre Historia de Usuario: <i>Realizar votación.</i>
Nombre de la persona que realiza la prueba: <i>Raynel Batistas Téllez</i>	
Descripción de la Prueba: <i>La página inicial del subsistema contiene a la izquierda una columna con el vínculo principal al subsistema CIGECI, en él se puede observar el link a la interfaz de votación, donde se encuentran todas las solicitudes que han sido revisadas. Se selecciona la solicitud a votar y se efectúa el clic en el botón realizar votación donde se encuentra la</i>	

<i>interfaz de votación y se realiza el voto con su debida observación, luego se presencia como el estado cambia a aceptado, rechazado o pendiente en dependencia de la decisión del votante.</i>
Condiciones de Ejecución: <i>El usuario debe tener identificado su rol de votante y al menos estar una solicitud en estado de revisada, además tiene que haber sido activada al menos una regla de votación.</i>
Entrada / Pasos de ejecución: <i>El votante escoge la opción realizar votación, señala la solicitud por la cual desea votar, ejecuta un clic en el botón realizar votación, escoge si va a ser rechazada, aceptada o se abstiene y presiona el botón aceptar.</i>
Resultado Esperado: <i>La votación se realizó correctamente obteniendo los resultados esperados según los requisitos previstos.</i>
Evaluación de la Prueba: <i>Satisfactoria</i>

Tabla 10. Prueba de aceptación de la H_U Realizar votación

3.6.3 Acta de Aceptación

A modo de concluir satisfactoriamente el trabajo se confeccionaron 2 actas de aceptación una dirigida al asesor de cooperación internacional de la facultad 3 y otra al Arquitecto principal del Grupo de Investigación y Desarrollo de la Plataforma SO3. En la misma consta la aceptación total por parte de los principales clientes del Subsistema para la gestión de los procesos de trámites de salidas y visitas, hacia y desde el exterior, en la Facultad 3 en su versión 2.0.

3.7 Conclusiones

Con el presente capítulo se obtuvo los resultados del diseño y las pruebas propuestas por la metodología. Arrojando el cumplimiento satisfactorio de la realización de las pruebas de unitarias y de aceptación. Mediante el transcurso de construcción, se encontraron errores que fueron rectificadas para el uso correcto del producto y la satisfacción del cliente. De esta manera logra el software la calidad adecuada para la utilización del producto.

CONCLUSIONES

En el diagnóstico inicial se puede constatar que la muestra seleccionada valora la necesidad e importancia de la realización del proceso de negocio de trámites de cooperación internacional, sin embargo presentan insuficiencias en la determinación de las variables que necesitan ser guardadas y manipuladas en determinado momento por los directivos y por tanto es necesario y novedoso el aporte brindado por el presente trabajo de diploma.

El Subsistema para la gestión de los procesos de trámites de salidas y visitas, hacia y desde el exterior, en La Facultad 3 que se presenta está caracterizado por su disponibilidad e integridad a través de una aplicación web.

La solución informática implementada, demostró el manejo eficiente de los datos relacionados con los trámites de salidas y visitas, hacia y desde el exterior, en La Facultad 3 y la integración con la plataforma tecnológica de esta, aspecto este medular, en la novedad e importancia del resultado científico alcanzado.

RECOMENDACIONES

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

- Implantar La Plataforma Informativa SO3 y a su vez el Subsistema para la gestión de los procesos de trámites de salidas y visitas, hacia y desde el exterior en su versión 2.0 para la Facultad 3.
- Estudiar los procesos asociados a la gestión de trámites de cooperación internacional con el fin de identificar nuevos procesos a integrar al subsistema
- Aplicar la solución en las diferentes facultades de la Universidad.
- Agregar un módulo de reporte que incluya como principales funcionalidades la exportación de pdf y la impresión de la información de las planillas obtenidas durante todo el proceso de cooperación internacional.
- Incluir una grafica representativa sobre cualquier tipo de reporte que se necesite en el subsistema, de manera que pueda observarse la información de forma visual.

BIBLIOGRAFÍA

1. **informatica, Universidad de las ciencias.** Entorno virtual de aprendizaje. [En línea] 2012.
2. [En línea] <http://definicion.de/gestion/>.
3. **Lugo, Bernardo Mendez.** [En línea] www.southsouth.info/profiles/blogs/concepto-de-cooperacion.
4. **Oscar Jaramillo.** [En línea] <http://www.cie.unam.mx/~ojs/pub/Termodinamica/node12.html>.
5. **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.
6. **Ayala, Reisser Mejias y Ruiz, Keytia Quintero.** *Sistema Informatizado de Cooperación Internacional. Análisis y Diseño del Subsistema de Trámites.* La Habana : s.n., 2008.
7. **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.
8. **Mora, Sergio Lujan.** Programacion de Aplicaciones Web: historia, principios básicos y clientes web.
9. *SXP Metodología ágil para proyectos de software libre.* **Abad, Ing. Abel Meneses.** 2009.
10. **Romero, Oscar Casasola.** Programacion en Castellano. 2010.
11. **Pressman, Roger S.** Ingeniería del Software. Un enfoque práctico. 2004.
12. Visual Paradim. [En línea] 24 de enero de 2011. www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p.
13. EcuRed. [En línea] 15 de 2 de 2013. http://www.ecured.cu/index.php/Sistema_Gestor_de_Base_de_Datos.
14. **Toledo, Albaro.** UPTODOWN. [En línea] 21 de 1 de 2013. <http://postgresql.uptodown.com>.
15. Guia Ubuntu. [En línea] [Citado el: 24 de enero de 2012.] http://www.guia-ubuntu.org/index.php?title=PgAdmin_III..
16. **Alvarez, Miguel angel.** desarrolloweb. [En línea] 2009.
17. **Maestros del web.** Maestros del web. [En línea] 2010. [Citado el: 8 de febrero de 2011.] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.

18. Oficina España. [En línea] [Citado el: 29 de enero de 2013.] <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>.
19. Desarrollo Web. [En línea] [Citado el: 29 de enero de 2013.] <http://www.desarrolloweb.com/articulos/que-es-html.html>.
20. **Patxi**. Eslomas.com. *Frameworks de Zend para el desarrollo de aplicaciones PHP*. [En línea] 3 de julio de 2007. [Citado el: 8 de febrero de 2011.] <http://www.eslomas.com/index.php/archives/2007/07/03/framework-de-zend-para-el-desarrollo-de-aplicaciones-php/>.
21. Libro JQuery. [En línea] <http://librojquery.com/>.
22. TESIS DE MÁSTER MÁSTER DE INVESTIGACIÓN EN INTELIGENCIA ARTIFICIAL. [En línea] http://oa.upm.es/1064/1/JOSE_MARIA_FONT_FERNANDEZ.pdf.
23. **Borja, Angie Alexandra**. Slideshare. [En línea] [Citado el: 3 de febrero de 2013.] <http://www.slideshare.net/inspilateunaula/mapeo-de-objeto-relacional>.
24. PHP. [En línea] php.net.
25. duplika. [En línea] [Citado el: 4 de febrero de 2013.] <http://www.duplika.com/blog/que-son-los-servidores-web-y-por-que-son-necesarios>.
26. AplicacionesEmpresariales. [En línea] [Citado el: 3 de febrero de 2013.] <http://www.aplicacionesempresariales.com/apache-el-servidor-web-mas-reconocido.html>.
27. **Peñalver Romero, Gladys Marsi, y otros**. *Metodología ágil para proyectos de software libre*. Ciudad de la habana : s.n., 2009.
28. Entorno vital de aprendizaje. [En línea] 2013. http://eva.uci.cu/file.php/160/Curso_2010-2011/Semana_9/Conferencia_7/Materiales_Basicos/Documentacion_sobre_Pruebas.pdf.
29. 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.]
30. **Pérez, Ing. Joisel Pérez**. *Línea base de la arquitectura de software Contabilidad*. s.l. : ERP Cuba, 2010.
31. **Dirección de Economía del MINFAR**. *Manual del Sistema de Contabilidad Financiera para la Actividad Presupuestada en las FAR*. Plaza de la Revolución, Ciudad de la Habana : Centro de Información para la Defensa del MINFAR, 2003.

32. **Microsoft.** Microsoft Dynamics|Soluciones empresariales ERP y CRM. [En línea] Microsoft Dynamics, 2011. [Citado el: 20 de enero de 2011.] <http://www.microsoft.com/dynamics/es/es/default.aspx>.
33. **Openbravo Wiki.** Openbravo Wiki. [En línea] 26 de 6 de 2009. [Citado el: 20 de enero de 2011.] http://wiki.openbravo.com/wiki/User_Manual_2.40/es.
34. **CENDITEL.** Plataforma de Desarrollo de Software Libre (PDSL). *Manual del Usuario del Sistema de Control de Versiones (SVN)*. [En línea] CENDITEL, 2011. [Citado el: 20 de enero de 2011.] <http://plataforma.cenditel.gob.ve/wiki/ManualUsuarioSvn>.
35. **BETSIME.** Revista BETSIME. *La revista del empresario cubano*. [En línea] 2011. [Citado el: 20 de enero de 2011.] http://www.betsime.disaic.cu/secciones/eco_enemar_07.htm#2.
36. **CITMATEL.** Rodas XXI Sistema Integral Económico Administrativo. [En línea] 2011. [Citado el: 20 de enero de 2011.] <http://www.rodasxxi.cu/contabilidad.php>.
37. **Tobías, María.** Lainformacion.com. [En línea] 8 de 12 de 2008. [Citado el: 20 de enero de 2011.] <http://tobiassig08.bitacorras.com/archivos/2008/12/08/software-empresarial-y-nuevas-tecnologias-erp2>.
38. **Scribd.** Scribd. [En línea] 2010. [Citado el: 22 de enero de 2011.] <http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.
39. **Romero, Oscar Casasola.** Programacion en Castellano. [En línea] 27 de agosto de 2010. [Citado el: 24 de enero de 2011.] http://www.programacion.com/articulo/introduccion_a_uml_181.
40. **The PHP Group.** PHP: Hipertext Preprocesor. [En línea] php.net, 4 de 2 de 2011. [Citado el: 6 de febrero de 2011.] <http://php.net/manual/es/intro-what-is.php>.
41. **Colectivo de Json.** JSON. [En línea] JSON, 2011. [Citado el: 22 de enero de 2011.] <http://www.json.org/json-es.html>.
42. **Grupo de autores.** HTML.net. [En línea] 2010. [Citado el: 20 de enero de 2011.] <http://es.html.net/tutorials/css/lesson1.php>.
43. **Visual Paradim.** freedownloadmanager.org. [En línea] freedownloadmanager.org, 2008. [Citado el: 24 de enero de 2011.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
44. **Toledo, Álvaro.** UPtoDOWN. [En línea] 2010. [Citado el: 24 de 1 de 2011.] <http://postgresql.uptodown.com/>.

45. **Guía Ubuntu.** Guía Ubuntu. [En línea] 10 de marzo de 2008. [Citado el: 24 de enero de 2011.] http://www.guia-ubuntu.org/index.php?title=PgAdmin_III.
46. **IDE.** Wikipedia. [En línea] 2 de febrero de 2011. [Citado el: 6 de febrero de 2011.] http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado.
47. **Grupo editorial.** Entrebits.CL Comunidad Informática. [En línea] 10 de abril de 2008. [Citado el: 8 de febrero de 2011.] <http://www.entrebits.cl/foros/herramientas-de-programacion/11472-zend-studio-eclipse.html>.
48. **ProgramacionWeb.net.** ProgramacionWeb.net. [En línea] ProgramacionWeb.net, 2009. [Citado el: 8 de febrero de 2011.] <http://www.programacionweb.net/cursos/curso.php?num=2>.
49. **Colectivo editorial.** W3C Consortium. [En línea] 9 de enero de 2008. [Citado el: 8 de febrero de 2011.] <http://www.w3c.es/divulgacion/guiasbreves/tecnologiasxml>.
50. **machado, Daymel.** *Implementación del módulo Distribución de productos*. Ciudad de la Habana : s.n., 2010.
51. **Ciberaula.** Ciberaula.com. *Una Introducción a APACHE*. [En línea] 2010. [Citado el: 8 de febrero de 2011.] http://linux.ciberaula.com/articulo/linux_apache_intro/.
52. **Doctrine-Project.org.** Doctrine. [En línea] 2010. [Citado el: 8 de febrero de 2011.] <http://www.doctrine-project.org>.
53. **Vargas.** Wordpress.com. *Librería ExtJs*. [En línea] 6 de agosto de 2007. [Citado el: 8 de febrero de 2011.]
54. **Mozilla Firefox.** GetFirefox. [En línea] 2009. [Citado el: 6 de febrero de 2011.] <http://www.getfirefox.es/firefox-features>.
55. **Colectivo de editores.** Programación I y Estructura de Datos. [En línea] 2007-2008. [Citado el: 6 de febrero de 2011.] <http://upsg01.foroactivo.com/f10-tema-2-estructura-de-datos-paralelo-22>.
56. Wikipedia. [En línea] 16 de febrero de 2011. [Citado el: 30 de abril de 2011.] http://es.wikipedia.org/wiki/Modelo_de_datos.
57. Sparx Systems. [En línea] Sparx Systems Pty Ltd, 2007. [Citado el: 5 de mayo de 2011.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.
58. **Hernandez, Juan Darien Macias.** Dragones, Facultad 3. [En línea] Proyecto PCI, Facultad 3., 15 de marzo de 2011. [Citado el: 20 de abril de 2011.] http://facultad3.uci.cu/index.php?option=com_content&view=article&id=331&Itemid=146.
59. **ERP Cuba.** *Vista de datos de la arquitectura* . 2010.

60. **Fernández González, Mairelys y Zorrilla Rivera, Osley.** Diseño e implementación del componente Ajuste al Costo del Subsistema Costos y Procesos del Sistema Integral de Gestión de Entidades CEDRUX. *Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.* Ciudad de la Habana : UCI, 2010.
61. **Martinez, Rafael.** PostgreSQL-es. [En línea] PostgreSQL-es, 6 de junio de 2009. [Citado el: 24 de mayo de 2011.] <http://www.postgresql.org.es/node/297>.
62. **Bravo, Alfredo Perez.** Cooperacion Tecnica Internacional. 2007.
63. **ERP-Cuba, Proyecto.** *Modelo de Desarrollo Orientado a Componentes del proyecto ERP-Cuba.* 2008.
64. **International, V.P. 2006.** Introducción a los sistemas y herramientas CASE. 2006.
65. **GARCIA, X.C and J. M. ALFONSO.** Introducción a los Sistemas de Gestión de Contenidos. 2004.
66. **Stephen A. White, IBM Corporation.** Introduction to BPMN.
67. **Potencier, Fabien.** *Symfony la guia definitiva .* 2008.
68. **Academia, Real.** *diccionario de la lengua española.* 2007.
69. **Guzmán., Dra.C. Esther María Pino.** *La Excelencia en la Dirección Educativa Latinoamericana.* 2007.
70. **Vargas.** Libreria Extjs. [En línea] Wordpress.com.
71. **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>.
72. **Unidad de Compatibilización Integración y Desarrollo.** *Proceso de Desarrollo y Gestión de Proyectos de Software.* Ciudad de la Habana : s.n., 2009.