Universidad de las Ciencias Informáticas

Facultad 6

Sistema para la gestión del proceso de impresiones del Vicedecanato de Administración de la facultad 6 de la Universidad de las Ciencias Informáticas.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

Autores: Osviel Rodríguez Valdés

Bernardo Hernández González

Tutores: MSc. Omar Mar Cornelio

Ing. Flavio Roche Rodríguez

La Habana, junio de 2015

"Año 57 de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaramos ser autores del trabajo ***"Sistema para la gestión del proceso de impresiones del Vicedecanato de Administración de la facultad 6 de la Universidad de las Ciencias Informáticas"*** y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Bernardo Hernández González

_____

Firma del Autor

Osviel Rodríguez Valdés

_____

Firma del Autor

Msc. Omar Mar Cornelio

_____

Firma del Tutor

Ing. Flavio Roche Rodríguez

_____

Firma del Tutor

DATOS DE CONTACTO

Autor:
Osviel Rodríguez Valdés
Universidad de las Ciencias Informáticas
Correo electrónico: orguez@estudiantes.uci.cu

Autor:
Bernardo Hernández González
Universidad de las Ciencias Informáticas
Correo electrónico: bhernandez@estudiantes.uci.cu

Tutor:
Msc. Omar Mar Cornelio
Universidad de las Ciencias Informáticas
Correo electrónico: omarmar@uci.cu

Tutor:
Ing. Flavio Enrique Roche Rodríguez
Universidad de las Ciencias Informáticas
Correo electrónico: feroche@uci.cu

DEDICATORIA

A mi mamá Ofelia, por todo el esfuerzo y la dedicación para formar al hombre en el que me he convertido y por ser la guía de cada uno de mis logros.

A mi padre, por apoyarme incondicionalmente, en los buenos y malos momentos.

A toda mi familia, mis abuelos, mis tíos, mis primos, mi padrastro, que en todos estos años de estudio han estado a mi lado aconsejándome y formándome en cada paso que dado.

A mi novia Dayana, por toda la entrega, el apoyo y el tiempo que me ha dedicado.

A mis amigos, mis profesores y a todos los que de una manera u otra han contribuido con mi preparación y formación como profesional.

BERNARDO

A mis padres: María Esther, María Alayón, Osvaldo Rodríguez, Bernardo Fernández e Ibis Elena de los que siempre he recibido apoyo incondicional, en especial a mi mamá la cual ha sido el motor impulsor de mi desempeño.

A mis hermanos que quiero con la vida y para los cuales siempre he tratado de ser un ejemplo. Aún les quedan retos importantes en la vida, siempre podrán contar conmigo, para ellos también es este trabajo.

A Lionel, por su entrega incondicional, su paciencia y apoyo en los momentos más difíciles cuando la docencia y la FEU robaban todo mi tiempo, su compañía ha sido clave en mi felicidad.

A mis amigos, los que han servido de soporte en las buenas y malas, en los que siempre he podido confiar y los que siempre han esperado más de mí.

A mis profesores, cuya experiencia, sabiduría y conocimientos hay forjado mi carácter y contribuido a mi formación profesional.

OSVIEL

RESUMEN

Con el auge de las nuevas tecnologías de información y comunicaciones, diversos procesos que se llevan a cabo en las organizaciones se han favorecido sustancialmente, al punto de llegar a convertirse estas nuevas alternativas en un factor determinante en la eficiencia y organización. En el Vicedecanato de Administración de la facultad 6 de la Universidad de las Ciencias Informáticas, se desarrolla el proceso de impresiones, en el cual se han manifestado varias deficiencias como continuas interrupciones e insuficiente control y aprovechamiento de recursos, que en su conjunto influyen negativamente en el funcionamiento de la mayoría de las tareas que se llevan a cabo en esta área. La presente investigación tiene como objetivo realizar el análisis, diseño, implementación y evaluación de un sistema informático, que contribuya al control de los recursos y la organización del proceso de gestión de impresiones, para favorecer la eficiencia, calidad y el apoyo a la toma de decisiones. Para llevar a cabo la propuesta de solución se utilizó la metodología de desarrollo OpenUP y como principal tecnología de desarrollo el *framework* Symfony 2.3, haciendo uso del lenguaje de programación PHP en su versión 5.5.

PALABRAS CLAVES: Gestión de impresiones, aplicación informática, toma de decisiones.

ABSTRACT

With the rise of new information and communications technologies, many processes that taking place in organizations have substantially favored, to the point of these new alternatives become a determining factor in the efficiency and organization. In the Management Vicedean of the Faculty 6 at the Informatics Sciences University, is carried out the print process, in which various shortcomings have emerged as continuous interruptions and insufficient control and use of resources, which together affect negatively the performance of this and most of the tasks carried out in this area. This research aims at the analysis, design, implementation and evaluation of a computer system that contributes to the control of resources and the organization of the prints management process, pretending promote efficiency, quality and support for decision-making in this activity. To carry out the proposed solution was defined use the OpenUP development methodology and as the main development technology the framework Symfony 2.3 using the PHP programming language in version 5.5.

KEYWORDS: Print management, computer application, decision-making.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

A partir del avance de las Tecnologías de Información y Comunicaciones (TIC), la sociedad ha visto nacer una era marcada por nuevas tendencias y mecanismos en el desarrollo de sus procesos. A nivel global, el manejo de la información se ha convertido en el centro de la toma de decisiones, y su correcta utilización es capaz de definir en la actualidad una posición importante en el ámbito competitivo de la industria, el mercado y los servicios. En Cuba, la búsqueda de nuevas alternativas que posibiliten una gestión eficiente de los grandes volúmenes de información que se generan, a partir de las ventajas que ofrecen las nuevas tecnologías, también ha sido una marcada tendencia, basada en el proceso de informatización de la sociedad.

La necesidad de perpetuar parte de la información que se genera en las organizaciones asociada a procesos esenciales y hasta cierto punto legal u oficial, se hace cada vez más presente. Las ventajas que con el paso del tiempo ha demostrado tener el almacenamiento de la información de manera impresa, constituye en la actualidad una de las garantías que sustentan su utilización, aún en una era en la que el uso de las nuevas tecnologías y los distintos medios de almacenamiento digital forman parte de la cotidianidad.

No son pocas las organizaciones que utilizan la impresión de documentos para almacenar información en el desarrollo de sus procesos, un ejemplo de estas es la Universidad de las Ciencias Informáticas (UCI), cuya estructura está formada por 7 facultades docentes y varias áreas administrativas. Cada facultad cuenta con varias direcciones o vicedecanatos, en los que desarrollan una serie de actividades independientes que contribuyen al funcionamiento general de la universidad.

En la facultad 6, específicamente en el Vicedecanato de Administración (VDA) se realiza de manera central el proceso de impresiones de la facultad. Para desarrollar esta tarea existen varias impresoras, atendidas por el asistente del vicedecano, el cual tramita todas las solicitudes de las diferentes áreas. Cada trabajador o estudiante autorizado se presenta en el local, con la información a imprimir en un dispositivo de almacenamiento y hace su solicitud al asistente, de manera personal. Como medida administrativa se estableció un horario de impresión para evitar interrupciones en el flujo de tareas diarias. El establecimiento de este horario no eliminó las interrupciones constantes y aún persisten las siguientes irregularidades:

- Existe personal en la oficina realizando solicitudes de impresión en cualquier momento del día, influyendo negativamente en el aprovechamiento de la jornada laboral y en el cumplimiento de los procesos administrativos.

- Se evidencia la carencia de un mecanismo que priorice la impresión de documentos importantes, lo que provoca que en ocasiones se impriman informaciones irrelevantes y no se aprovechen los recursos en función de cumplir los objetivos de la organización.

- El proceso actual no cuenta con un mecanismo de control sobre los recursos asignados para el proceso de impresiones.

- No se cuenta con un repositorio de impresiones que permita auditar la documentación impresa en intervalos de tiempos específicos, con el objetivo de constatar la eficiencia en el aprovechamiento de los recursos.

- Todo el proceso de impresión recae sobre el VDA, anulándose la responsabilidad directa que tienen los jefes de departamentos, jefes de áreas y demás miembros del consejo de dirección.

A partir de la problemática planteada se define como **problema de investigación**: ¿Cómo contribuir a la gestión de las impresiones en el Vicedecanato de Administración de la Facultad 6 de la UCI?

La investigación tiene como **objeto de estudio**: los sistemas para la gestión de información, enmarcado en el **campo de acción**: los sistemas para la gestión de impresiones.

Como **objetivo general** se plantea: desarrollar un sistema para la gestión del proceso de impresiones en el Vicedecanato de Administración de la Facultad 6 de la UCI.

Para guiar el proceso investigativo se plantean las siguientes **preguntas científicas**:

- ¿Cuáles son los principales fundamentos teóricos metodológicos relacionados con los sistemas de gestión de impresiones?
- ¿Qué técnicas, tecnologías y herramientas de desarrollo se podrían utilizar en la implementación de la propuesta de solución?

- ¿Qué técnicas de pruebas se pueden utilizar en la validación de la propuesta de solución y cómo se deben aplicar?

Para dar cumplimiento al objetivo general se definen las siguientes **tareas de investigación**:

- Elaborar el marco teórico referencial relacionado con los sistemas para la gestión de información de impresiones en red.
- Seleccionar las principales tecnologías y herramientas que se ajustan a la situación real del negocio para el desarrollo de la propuesta de solución.
- Elaborar el diseño del sistema para gestionar las impresiones desde la red.
- Realizar la implementación del sistema para gestionar las impresiones desde la red.
- Aplicar pruebas de funcionalidad al sistema para gestionar las impresiones desde la red, para detectar posibles no conformidades.

Para la realización de todo el proceso investigativo se utilizaron varios **métodos científicos**, agrupados en **teóricos y empíricos**.

Los métodos teóricos se utilizan en la adopción de la teoría científica y en el enfoque general para abordar los problemas científicos, posibilitando la interpretación conceptual de los datos empíricos y la explicación de los hechos. Están presentes en los diversos momentos del proceso de investigación y durante el análisis y comprensión de la información utilizada para el proceso de desarrollo (HERNÁNDEZ and COELLO 2011).

Los métodos teóricos presentes en esta investigación son:

Analítico – sintético: posibilitó la realización de un análisis acerca de las distintas soluciones existentes que facilitan la gestión de impresiones desde la red, para determinar sus principales ventajas y características, los principales conceptos asociados a este marco referencial y la selección de la metodología de desarrollo más apropiada para la propuesta de solución.

Los métodos empíricos por su parte, permiten estudiar los fenómenos observables directamente y contribuyen a confirmar las teorías (HERNÁNDEZ and COELLO 2011). En la presente investigación como método empírico se utilizó:

Entrevista: se aplicó a trabajadores de la dirección del VDA con el objetivo de constatar la existencia de dificultades en el proceso de la gestión de impresiones (Ver Anexo 1). Este método también se aplicó con el objetivo de conocer los detalles y el estado actual de la solución desarrollada para la Dirección de Información (Ver Anexo 2).

La **estructura del trabajo** quedó constituida en tres capítulos, conclusiones, anexos y bibliografía.

Capítulo 1: ***"Fundamentos Teóricos"***

En este capítulo se exponen los principales conceptos asociados a la propuesta de solución. Además se realiza un análisis de las principales soluciones existentes, con el objetivo de identificar las tendencias y las características de estas que pudieran aportar al diseño de la solución. Finalmente se define la metodología de desarrollo de software y las principales tecnologías y herramientas para su implementación y despliegue.

Capítulo2: ***"Análisis y diseño de la solución propuesta"***

En este capítulo se realiza el diseño del modelo de negocio, el cual constituye el punto de partida para el desarrollo de la propuesta de solución y se definen sus requisitos funcionales y no funcionales. Además se desarrolla el modelo de casos de uso, donde se exponen también los diferentes patrones utilizados y se especifican los elementos fundamentales de la arquitectura y el diseño, así como determinados artefactos que generan según la metodología de desarrollo definida.

Capítulo3: ***"Implementación y pruebas"***

En este capítulo se describe el proceso de implementación de la propuesta de solución y se diseña el diagrama de componentes. Además se especifican las pruebas realizadas sobre la solución y sus resultados, con el objetivo de determinar no conformidades, erradicarlas y probar que el sistema satisface sus requisitos funcionales y no funcionales.

CAPÍTULO 1: "FUNDAMENTOS TEÓRICOS"

Introducción

En este capítulo se presenta una descripción acerca de los principales conceptos asociados al marco referencial de la investigación. Se realiza un análisis acerca de las principales características y ventajas de las diferentes soluciones existentes que facilitan la gestión de impresiones y se realiza una comparación entre estas. Además, se fundamentan las herramientas, las tecnologías y la metodología a utilizar para el desarrollo de la propuesta de solución.

1.1 Principales conceptos asociados a la investigación

A continuación se exponen los principales conceptos relacionados con el marco teórico de la investigación, con el objetivo de profundizar en los distintos puntos de vista y asumir una posición al respecto.

1.1.1 Sistema de Información

Un sistema de información constituye un conjunto de elementos interrelacionados con el propósito de prestar atención a las demandas de información de una organización, para elevar el nivel de conocimientos que permitan un mejor apoyo a la toma de decisiones y desarrollo de acciones (DANGEL 2009). Según la Resolución 60 del año 2011 de la Contraloría General de la República de Cuba, referente a las normas de control interno, un sistema de información está constituido por los métodos y procedimientos establecidos para registrar, procesar, resumir e informar sobre las operaciones de una entidad. La calidad de la información que brinda el sistema afecta la capacidad de los directivos y ejecutivos para adoptar decisiones adecuadas que permitan controlar las actividades de la entidad (CUBA 2011).

Coincidiendo con lo planteado en los anteriores puntos de vista, en la presente investigación se define como Sistema de Información, al conjunto de elementos, métodos y procedimientos establecidos, que se interrelacionan con el objetivo de satisfacer las necesidades de una organización vinculadas al procesamiento de información, y que puede tener una influencia directa en los procesos de control y en la toma de decisiones.

1.1.2 Gestión de información en las organizaciones

Estrechamente relacionada con los sistemas de información, se encuentra la gestión de información en las organizaciones. Son varios los conceptos que existen para describir este término, Lynda Woodman plantea que la gestión de información es todo lo relacionado con obtener la información correcta, en la forma adecuada, para la persona indicada, al costo correcto, en el momento oportuno, en el lugar indicado para tomar la acción precisa (WOODMAN 1985), por su parte White la denomina como la coordinación eficiente y eficaz de la información procedente de fuentes internas y externas (WHITE 1985).

La Dra. Gloria Ponjuán define que cuando se menciona gestión de información se refiere a la gestión que se desarrolla en un Sistema de Información, si se trata de que el sistema tenga como propósito obtener salidas informacionales. Además la define como el proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económicos, físicos, humanos, materiales), para manejar información dentro y para la sociedad a la que sirve. Tiene como elemento básico la gestión del ciclo de vida de este recurso y ocurre en cualquier organización. Es propia también de unidades especializadas que manejan este recurso en forma intensiva, llamadas unidades de información (PONJUAN 1998).

A partir del análisis de los conceptos expuestos anteriormente, en la presente investigación se entiende por gestión de información al adecuado proceso de obtención, análisis y utilización de la información que fluye desde el ámbito interno o externo a una organización; y que puede influir notablemente en la toma de decisiones y en la eficiencia de los productos o servicios que esta genera.

1.1.3 Almacenamiento y procesamiento de la información en la UCI

El almacenamiento y procesamiento de la información tiene una gran importancia en la gestión de la información en las organizaciones, en la UCI este proceso generalmente se realiza a través del almacenamiento digital y el almacenamiento en formato duro o impreso, los cuales a continuación se detallan.

- **Almacenamiento digital**

Se define como almacenamiento digital al proceso en el que mediante documentos o archivos de diferentes formatos se almacena la información, generalmente en discos magnéticos o duros, discos

ópticos, memorias flash, unidades de estado sólido o en dispositivos virtuales como la computación en la nube, paradigma que figura como una de las tendencias que más auge ha cobrado en la actualidad.

Dentro de los principales formatos en que se almacena la información en documentos digitales, se destacan el DOC y XLS, desarrollados por Microsoft como formatos nativos de sus herramientas Microsoft Word (MICROSOFT 2011b) y Microsoft Excel (MICROSOFT 2011a) respectivamente, para su paquete ofimático privativo Office.

Otro comúnmente utilizado es el Formato de Documento Portable, PDF por sus siglas en inglés, que tiene como principal ventaja su característica de ser multiplataforma e independiente del software y el hardware, permitiendo ser usado en los principales sistemas operativos sin que se modifique ni el aspecto ni la estructura del documento original (ADOBE 2014).

Finalmente también se deben mencionar el ODT y el ODS, formatos para el almacenamiento de textos y hojas de cálculo respectivamente, de *Open Document*, que tienen como característica fundamental el ser formatos de archivo abierto y estándar para el almacenamiento de documentos ofimáticos (WITTMANN 2012).

El almacenamiento digital de documentos tiene sus ventajas, como por ejemplo el hecho de que un documento digital se puede compartir en la red y ser accedido simultáneamente por varios usuarios, puede ser duplicado y enviado a través de medios digitales, no se deterioran con el paso del tiempo y disminuyen el riesgo de pérdida de información ante incendios, robos, inundaciones u otros fenómenos. Como desventajas se debe mencionar que la longevidad de los soportes de almacenamiento que se utilizan es corta, si se comparan con la del papel de calidad, bien resguardado, además de volverse obsoleta con relativa rapidez, provocando que con frecuencia la compatibilidad hacia atrás sea limitada.

- **Almacenamiento en formato duro o impreso**

A pesar de las ventajas que provee el almacenamiento digital de documentos, existen varios procedimientos, incluso algunos amparados en marcos legales, que requieren la información almacenada en documentos de manera impresa, en ocasiones además con disposiciones oficiales para su tramitación. Ejemplo de estos casos son el procedimiento para solicitud de autorización de medios tecnológicos, la documentación de procesos disciplinarios, los numerosos modelos oficiales destinados al control de los

recursos, entre otros procesos que coinciden en su importancia para el desarrollo de las actividades y que aún para llevarse a cabo necesitan contar con evidencia impresa de los documentos, que pueda ser accesible en cualquier momento y bajo cualquier circunstancia.

1.1.4 Sistema de Gestión de Información

Con el avance de las TIC, han surgido novedosas alternativas que proveen varias ventajas a la hora de realizar la gestión de la información en las organizaciones. A partir de este nuevo escenario han cobrado auge los sistemas de gestión de información, de los cuales existen varios puntos de vista. Davis y Olson lo definen como un sistema integrado y automatizado para proveer la información que sostenga las funciones de operatividad, gestión y toma de decisiones en una organización (DAVIS and OLSON 1985).

Moreiro González define un sistema de gestión de información como el conjunto de políticas y normas relacionadas entre sí, que se establecen para el acceso y tratamiento de los recursos de información, incluye los registros administrativos, los archivos, el soporte tecnológico de los recursos y el público al que se destina (MOREIRO 1998).

En esta investigación se entenderá como Sistemas de Gestión de Información a un grupo de elementos informáticos y de recursos humanos que interactúan entre sí, para de conjunto facilitar los procesos de gestión de información de una organización y que como ventajas principales se puede mencionar un notable ahorro de recursos y de esfuerzos, facilidad en el manejo del proceso de gestión en particular y la posibilidad de obtener reportes automatizados más confiables, que permitirán un mejor proceso de control interno en las organizaciones y una mejor toma de decisiones, incrementando con ello la eficiencia.

1.2 Soluciones existentes que facilitan la gestión de impresiones

A continuación se exponen algunas de las soluciones existentes que facilitan la gestión de impresiones, tanto en el ámbito internacional como nacional. Posteriormente se realizará una comparación para definir características que pudieran aportar de alguna manera al desarrollo de la propuesta de solución.

1.2.1 Sistema de Impresión Centralizado DALÍ

Desarrollado por la Universidad de Murcia (UMU), este sistema de gestión de impresión permite realizar un uso más eficiente y cómodo de los recursos de impresión, así como supervisar y moderar el uso de los mismos.

El sistema cubre todo el colectivo universitario incluyendo alumnos, personal docente y trabajadores. Toda la impresión de la Universidad está gestionada por el sistema *Dalí* de forma que el envío de trabajos tiene que hacerse siempre de forma autenticada utilizando la cuenta de correo de la UMU (@um.es).

La impresión se puede realizar de dos formas:

- **Impresión directa:** los trabajos de impresión se envían a una impresora específica, en cuyo caso la impresión es inmediata.
- **Impresión indirecta:** los trabajos de impresión se enviarán a una cola genérica, quedando pendientes de ser liberados. Para liberarlos deberá acudir a una estación de liberación, un ordenador junto a las impresoras. Autenticándose con usuario de correo UMU y contraseña, podrá liberar el/los trabajos de impresión (MORENO 2014).

1.2.2 CZ Print Job Tracker

CZ Print Job Tracker es un potente gestor monitor de impresión centralizado para entornos de impresión Windows, Linux, Unix, Mac, tanto con servidor de impresión de Windows como sin él. Tiene como principal característica que con solo instalarlo en un equipo se puede controlar, monitorizar, contar y administrar la impresión de todos los otros equipos conectados a la misma red.

Este sistema permite además gestionar, limitar y contar las impresiones, así como supervisar y llevar un registro de todas las actividades de impresión, analizar y monitorizar los costes, facturar los trabajos indicando los códigos del cliente o del proyecto en la estación de trabajo, confirmar y autentificar los trabajos en la estación antes de su impresión efectiva, eliminar el desperdicio de papel y reducir el tiempo de mantenimiento de la impresora (SKRABA 2013).

1.2.3 Google Cloud Print

Google Cloud Print es una nueva tecnología que conecta impresoras a la Web para que estén disponibles desde los dispositivos de mayor uso frecuente como son el teléfono, una tableta, una computadora o cualquier otro conectado a la Web desde el cual se necesite imprimir, desde aquí este sistema enruta los trabajos de impresión y los envía a la impresora conectada a Internet.

Este servicio se integra con las versiones móviles de *Gmail* y *Google Docs*, lo que permite a los usuarios imprimir desde sus dispositivos móviles mediante una opción de impresora en la página vista preliminar del navegador web *Google Chrome* desde su versión 16 en hasta las más actualizadas. Realiza la impresión de manera directa y necesita una infraestructura tecnológica compatible u optimizada para su funcionamiento (SARASWAT 2014).

1.2.4 Sistema de Gestión de Impresiones de la Dirección de Información de la UCI

A partir de entrevistas realizadas por los autores se pudo constatar que en el año 2009, se realizó una aplicación web para la gestión de impresiones de la Dirección de Información de la UCI. Esta aplicación permitía gestionar las impresiones que tenían asignadas en aquel entonces las brigadas de estudiantes y hasta cierto punto solucionaba parte de la problemática existente en la facultad 6. Este sistema en la actualidad no existe, pues el procedimiento específico para el que desarrolló ya no se implementa, sus desarrolladores ya no se encuentran en la UCI para brindarle soporte y la estación de trabajo donde estaba instalado sufrió baja por dificultades técnicas, con lo cual se perdió toda su información. Este sistema solo permitía asignar una cantidad de hojas determinada a una brigada y luego ir descontando a medida que se solicitaba un nuevo servicio (ALFONSO and GARCÍA 2009).

Una vez analizadas cada una de las soluciones existentes planteadas se procede a realizar una comparación entre ellas, con el objetivo de determinar si pueden servir para dar solución a la problemática planteada o pueden aportar algo que contribuya en la realización de un nuevo sistema.

Tabla. 1 Comparación entre las soluciones existentes que facilitan la gestión de impresiones.

Solución	Flujo de impresión	Propietario	Plataforma	Reportes	Soporte

DALI	Directa e Indirecta	Si	Multiplataforma	No	Si
CZ Print Job Tracker	Directa	Si	Multiplataforma	Si	Si
Google Cloud Print	Directa	Si	Multiplataforma	No	Si
Sistema de Gestión de Impresiones de la Dirección de Información de la UCI	Indirecta	No	Multiplataforma	No	No

Al concluir la comparación y su posterior análisis se constató que las soluciones existentes, aunque presentan funcionalidades útiles para el proceso de impresión de la facultad 6 no satisfacen todas las necesidades ya que de forma general:

- Están hechas para ser utilizadas en impresoras de alta tecnología, que activan sus sistemas mediante tarjetas magnéticas y/o huellas digitales. Estas impresoras inteligentes son muy costosas y no se cuenta con ningún ejemplar en la facultad.
- No tienen mecanismos de asignación y control de recursos, función primordial que exige hoy el VDA.
- En la mayoría de los sistemas, el flujo de impresión es no asistido y por políticas de la universidad el asistente del VDA es responsable del equipo y del contenido legal de la información impresa.

Dentro de las principales funcionalidades que se consideran útiles y que de manera independiente presentan estas soluciones, se identificaron las siguientes:

- La capacidad de generar y exportar reportes del comportamiento del proceso de impresiones que provee el CZ Print Job Tracker.
- El sistema de autenticación mediante la cuenta de correo institucional, que presenta el sistema DALI en la Universidad de Murcia.
- El flujo de impresión asistido o indirecto que provee el Sistema de Impresiones de la Dirección de Información y el sistema DALI.

- La característica que presentan todos estos sistemas de funcionar en sistemas operativos de múltiples plataformas.

A partir de los análisis realizados se concluye con la necesidad de desarrollar un nuevo sistema para informatizar el proceso de gestión de impresiones del VDA de la facultad 6. La solución propuesta debe permitir un flujo de impresión indirecto o asistido, debido a que por disposiciones oficiales reguladas en las resoluciones 60 del 2011 de la Contraloría General de la República de Cuba (CGR), referente al control interno, y 127 del 2007 del Ministerio de Comunicaciones de Cuba (MIC), así como en las normas y políticas de uso de los medios tecnológicos de la UCI, el asistente encargado de la estación de trabajo en que se encuentra la impresora, es el responsable de la seguridad y legalidad de la información que se imprime en esta. Además la solución debe ser multiplataforma, desarrollarse sobre la base del software libre y debe permitir la visualización de estadísticas que permitan identificar tendencias o comportamientos relacionados con el uso de los recursos, para contribuir al mejor control de los mismos.

1.3 Metodología de desarrollo de software

Se entiende por metodología de desarrollo de software a una colección de pasos a seguir los cuales guían el trabajo referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. Una metodología define con precisión los artefactos, roles y actividades presentes en el proceso de desarrollo de software. La finalidad de una metodología de desarrollo es garantizar la calidad del trabajo, logrando con ello el cumplimiento de los requisitos iniciales, minimizando las pérdidas de tiempo en el proceso de generación de software.

No existe una metodología de software universal. El hecho de que cada software presente necesidades y entornos diferentes, que deben tenerse en consideración, hace que el proceso deba ser adaptable a sus características. Por otra parte cada metodología tiene características propias del ambiente de desarrollo, constituyendo estos los principales aspectos que rigen el proceso de selección de la metodología adecuada para la construcción de un software (JACOBSON *et al.* 2000).

Las metodologías de desarrollo de software, por sus características se pueden clasificar en tradicionales y ágiles.

Las metodologías tradicionales centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto que se define completamente en la fase inicial del proceso de desarrollo. Otras características importantes son el alto costo al implementar un cambio y al no ofrecer una buena solución para proyectos donde el entorno puede ser volátil, se considera a la arquitectura del software como esencial y se orienta hacia el trabajo en grandes grupos que pueden ser incluso distribuidos.

Por su parte las metodologías ágiles se basan en dos aspectos fundamentales, el retrasar las decisiones y la planificación adaptativa. Esto posibilita que la documentación necesaria para el proceso de desarrollo se elabore en etapas que pueden ser posteriores a la fase inicial y puedan ser adaptadas a cambios que se presenten en el proceso (FIGUEROA *et al.* 2011).

A partir del análisis de estos dos tipos de metodologías de desarrollo, se realiza la siguiente comparación entre ambas:

Tabla. 2 Comparación entre las metodologías de desarrollo de software.

Metodología Tradicional	Metodología Ágil
La arquitectura del software es esencial y se expresa mediante modelos.	Concede menos importancia a la arquitectura del software.
Se enfoca hacia el trabajo de grupos grandes que pueden incluso estar distribuidos.	Se enfoca hacia el trabajo en grupos pequeños.
El cliente no forma parte del equipo de desarrollo. Solamente interactúa con este a través de reuniones definidas u oficiales, en algunos momentos del proceso de desarrollo.	El cliente es parte del equipo de desarrollo e interactúa con este constantemente mediante reuniones que pueden ser hasta cierto punto informales.
Genera muchos artefactos y presenta numerosos roles en el equipo de desarrollo.	Genera pocos artefactos y presenta pocos roles en el equipo de desarrollo.

Presenta cierta resistencia a los cambios.	Está especialmente preparada para enfrentar cambios que surgen durante el desarrollo del proceso.

A partir de la comparación realizada se define para la propuesta de solución una metodología de clasificación ágil. Esta decisión se debe a que el equipo de desarrollo está formado por pocos miembros y el vicedecano administrativo, quien es considerado el cliente principal, tiene la disposición de interactuar con el equipo mediante reuniones en el momento en que sea necesario.

Las metodologías ágiles generan pocos artefactos y presentan pocos roles, lo que posibilita dedicar más tiempo al proceso de implementación y terminar la solución con mayor rapidez. Otro factor importante es que durante el proceso de desarrollo existe una importante probabilidad de que sucedan cambios en la concepción de algunos elementos relacionados con la propuesta de solución, situación para la cual las metodologías ágiles están especialmente preparadas.

Dentro de las metodologías ágiles más utilizadas y mejor documentadas está OpenUP *(Open Unified Process)*, la cual conserva las características principales del modelo de desarrollo de la metodología tradicional RUP *(Rational Unified Process),* incluye el desarrollo iterativo, permite identificar los requisitos operacionales del sistema, prever las interacciones con los usuarios y prevenir los posibles riesgos en el desarrollo del sistema.

OpenUP es una forma de desarrollo más ágil y ligera, consiste en equipos a los cuales se les asigna una fase del desarrollo que tienen que complementarse entre sí para obtener un buen producto final, no puede ser una sola persona la que realice todo el trabajo pues esto podría ocasionar que se pierdan de vista ciertas características importantes, por ejemplo para un proyecto pequeño se constituyen equipos de hasta seis personas e implican hasta seis meses de esfuerzo del desarrollo (TABARES *et al.* 2013).

La mayoría de los elementos de OpenUP están declarados para fomentar el intercambio de información entre los equipos de desarrollo y mantener un entendimiento compartido del proyecto, sus objetivos, alcance y avances.

OpenUp posee 4 principios básicos:

- Colaborar para sincronizar intereses y compartir conocimiento. Este principio promueve prácticas que impulsan un ambiente de equipo saludable, facilitan la colaboración y desarrollan un conocimiento compartido del proyecto.
- Centrarse en la arquitectura de forma temprana para minimizar el riesgo y organizar el desarrollo.
- Equilibrar las prioridades para maximizar el beneficio obtenido por los interesados en el proyecto. Este principio promueve prácticas que permiten a los participantes de los proyectos desarrollar una solución que maximice los beneficios obtenidos por los participantes y que cumple con los requisitos y restricciones del proyecto.
- Desarrollo evolutivo para obtener retroalimentación y mejoramiento continuo. Este principio promueve prácticas que permiten a los equipos de desarrollo obtener retroalimentación temprana y continua de los participantes del proyecto, permitiendo demostrarles incrementos progresivos en la funcionalidad (BALDUINO 2007).

Todo proyecto en OpenUP consta de cuatro fases: inicio, elaboración, construcción y transición. Cada una de estas fases se divide a su vez en iteraciones, lo cual tiene como ventaja que permite a los integrantes del equipo de desarrollo aportar con micro-incrementos, que pueden ser el resultado del trabajo de unas pocas horas o unos pocos días. Además este ciclo de vida provee a los clientes de una visión del proyecto, transparencia y los medios para que controlen la financiación, el riesgo, el ámbito y el valor de retorno esperado. A continuación se muestran los detalles de estas cuatro fases:

Fase de inicio:

En esta fase, las necesidades de cada participante del proyecto son tomadas en cuenta y plasmadas en objetivos del proyecto. Se definen para el proyecto: el ámbito, los límites, el criterio de aceptación, los casos de uso críticos, una estimación inicial del coste y un boceto de la planificación.

Fase de elaboración:

En esta fase se realizan tareas de análisis del dominio y definición de la arquitectura del sistema. Se debe elaborar un plan de proyecto, estableciendo unos requisitos y una arquitectura estable. Por otro lado, el proceso de desarrollo, las herramientas, la infraestructura a utilizar y el entorno de desarrollo también se especifican en detalle en esta fase. Al final de la fase se debe tener una definición clara y

precisa de los casos de uso, los actores, la arquitectura del sistema y un prototipo ejecutable de la misma.

Fase de construcción:

Todos los componentes y funcionalidades del sistema que falten por implementar son realizados, probados e integrados en esta fase. Los resultados obtenidos en forma de incrementos ejecutables deben ser desarrollados de la forma más rápida posible sin dejar de lado la calidad de lo desarrollado.

Fase de transición:

Esta fase corresponde a la introducción del producto en la comunidad de usuarios, cuando el producto está lo suficientemente maduro. La fase de la transición consta de las subfases de pruebas de versiones beta, pilotaje y capacitación de los usuarios finales y de los encargados del mantenimiento del sistema. En función de la respuesta obtenida por los usuarios puede ser necesario realizar cambios en las entregas finales o implementar alguna funcionalidad más (SALGADO *et al.* 2013).

En la presente investigación se define como metodología de desarrollo OpenUP, que como todas las metodologías ágiles evita la elaboración de documentación, diagramas e iteraciones, lo que se traduce en una disminución del trabajo. Además es muy apropiada para proyectos pequeños y de bajos recursos, como es el caso del presente equipo de desarrollo, ya que permite disminuir las probabilidades de fracaso e incrementar las probabilidades de éxito, mediante la detección de errores tempranos a partir de su ciclo iterativo, lo cual supone una ventaja importante en el desarrollo de la propuesta de solución.

1.4 Herramientas y tecnologías utilizadas

1.4.1 Lenguaje de modelado UML 2.0

Lenguaje Unificado de Modelado (UML) ha sido el estándar de la industria para visualizar, especificar, construir y documentar los artefactos de los sistemas software. Como lenguaje de modelado estándar de facto, UML favorece la comunicación y reduce la confusión entre los participantes de un proyecto software. La viabilidad y el ámbito del lenguaje han crecido con la reciente estandarización de UML 2.0. Su facilidad de uso permite a los usuarios modelar todo tipo de sistemas, desde sistemas de información de empresas y aplicaciones Web distribuidas hasta sistemas embebidos de tiempo real (BOOCH 2006).

UML es el resultado del trabajo realizado por Grady Booch, James Rumbaugh e Ivar Jacobson. Está constituido por un conjunto de diagramas y permite generar un anteproyecto de la propuesta de solución que permite tanto a los autores como a los clientes una mayor comprensión y claridad acerca de lo que debe hacer el sistema.

1.4.2 Herramienta de modelado Visual Paradigm 8.0

Para el modelado se utilizó Visual Paradigm para UML 8.0 por ser una herramienta CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador) que soporta el ciclo de vida completo en el desarrollo de software: análisis y desarrollo orientados a objetos, construcción, prueba y despliegue (PRESSMAN 2008). Entre las características fundamentales que determinaron su selección para el modelado durante todo el proceso de desarrollo del software están las siguientes:

- Apoya todo lo básico en cuanto a artefactos generados en las etapas de definición de requerimientos y de especificación de componentes.
- Brinda apoyo adicional en cuanto a generación de artefactos automáticamente.
- Genera modelos VP-UML instantáneamente a partir de código binario.
- Tiene disponibilidad en múltiples plataformas.
- Brinda la posibilidad de intercambiar información mediante la importación y exportación de ficheros.
- Permite la generación de código e ingeniería inversa, además de la generación de documentación.

1.4.3 Sistema Gestor de Bases de Datos (SGDB) PostgreSQL 9.3

PostgreSQL 9.3 es un SGBD objeto-relacional, distribuido bajo licencia BSD *(Berkeley Software Distribution)* y con su código fuente disponible libremente. Es el SGBD de código abierto más potente del mercado. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (DOUGLAS and DOUGLAS 2003).

Se define esta herramienta para el desarrollo de la propuesta de solución, debido a la estabilidad, potencia, robustez y facilidad de administración que provee. Además funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios, lo que se traduce en una garantía de eficiencia, dado que el sistema en desarrollo debe ser capaz de enfrentar múltiples conexiones simultáneas.

1.4.4 PgAdmin III 1.18

PgAdmin III es una aplicación de diseño y manejo de bases de datos para su uso con PostgreSQL. Se puede utilizar para manejar versiones de PostgreSQL a partir de la 7.3 y funciona sobre casi todas las plataformas. Es desarrollado por una comunidad de expertos alrededor del mundo y está disponible en más de una docena de idiomas publicado bajo la licencia de PostgreSQL.

Esta herramienta de software libre fue diseñada para responder a las necesidades de todos los usuarios, desde la escritura de simples consultas SQL *(Structured Query Language)* a la elaboración de bases de datos complejas. La interfaz gráfica es compatible con todas las características de PostgreSQL y facilita la administración (COULIBALY 2011).

La aplicación también incluye un editor de la sintaxis SQL, un editor de código del lado del servidor, un agente para la programación de tareas «SQL/batch/shell», soporte para el motor de replicación Slony-I y mucho más. La conexión del servidor se puede realizar mediante TCP/IP o *Unix Domain Sockets* y puede ser cifrado mediante SSL *(Secure Sockets Layer)* por seguridad. No se requieren controladores adicionales para comunicarse con la base de datos del servidor (COMUNITY 2012).

Dentro de las principales ventajas, que se tuvieron en cuenta para la selección de esta herramienta están su seguridad, así como su flexibilidad y capacidad de integración con bases de datos y con otros SGBD.

1.4.5 Servidor web Apache 2.2.22

Apache 2.2.22 es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos. Entre sus principales características se destacan que es multiplataforma, libre de costo, permite la configuración de ciertos módulos de programación, así como la restricción a determinados sitios web, conexiones seguras a través de SSL y configuración de servidores virtuales.

Dadas las características de Apache, se hace notable su amplia utilización en diversos proyectos, se tiene en cuenta el gran volumen de usuarios que acuden a sus servicios. Todo ello se fundamenta también en sus ventajosas opciones, reconocidas y registradas en las diferentes fuentes consultadas, dentro de las cuales se encuentran:

- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- Tiene una alta configurabilidad en la creación y gestión de logs.
- Es robusto y seguro.
- Es totalmente gratuito y se distribuye bajo la licencia *Apache Software License*, que permite la modificación del código.
- Funciona sobre todas las versiones recientes de Unix y Linux, Windows y BeOs.
- Admite una gran cantidad de lenguajes de script como Perl, PHP y Python. (GARCIA 2014).

1.4.6 Entorno de desarrollo integrado NetBeans 8.0

En la implementación de la propuesta de solución se utiliza el IDE *(Integrated Development Enviorment)* NetBeans 8.0 que es una herramienta de código abierto con una gran base de usuarios y una comunidad en constante crecimiento.

Es un producto libre, gratuito, sin restricciones de uso y con un número importante de módulos para extenderlo. Está compuesto por una base modular y extensible usada como una estructura de integración para crear grandes aplicaciones de escritorio. Entre sus características están las administraciones de ventanas, almacenamiento, interfaces y configuraciones de usuario. Además, empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones (NETBEANS 2015).

1.4.7 Lenguaje de programación PHP 5.5

PHP (*Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Puede emplearse en los sistemas operativos que más se utilizan, incluyendo Linux, numerosas variantes de Unix, Microsoft Windows, Mac OS X y RISC OS (PLEVA 2013).

Dentro de los elementos que definen la selección para su utilización en el desarrollo de la propuesta de solución, se puede mencionar que admite la mayoría de servidores web que se utilizan en la actualidad, incluyendo al Apache, que es el que se utilizará en el presente trabajo. Además cuenta con soporte para comunicarse con otros servicios usando el protocolo LDAP, que será el que se utilizará para la autenticación de los usuarios en el sistema.

Dentro de sus principales ventajas además se pueden mencionar:

- Está orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El soporte de sesiones en PHP consiste en una manera de guardar ciertos datos a través de diferentes accesos web. Esto permite crear aplicaciones más personalizadas y mejorar las características del sitio web.
- Utiliza las conexiones persistentes a las base de datos, con el objetivo de aumentar la eficiencia, en los casos en que las recargas para crear enlaces al servidor SQL son altas y soporta el manejo de excepciones.
- Su programación es segura y confiable, dado que el código PHP no es visible al navegador web y al cliente, pues es el servidor quien ejecuta el código y envía el resultado al navegador mediante código HTML (PLEVA 2013).

1.4.8 Lenguaje de programación JavaScript 1.8.3

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Es un lenguaje interpretado, por lo que no es necesario compilar los programas para ejecutarlos. Actualmente es la tecnología más extendida en el enriquecimiento de páginas web del lado del cliente (EGUILUZ 2006).

Este lenguaje será utilizado en la solución propuesta principalmente en el proceso de validación de datos entrados por los usuarios al sistema, ya que permite ejecutar instrucciones como respuesta a las acciones del usuario.

1.4.9 Framework de desarrollo Symfony 2.3

Un *framework* simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. También facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Symfony es un completo *framework* diseñado para optimizar el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Este *framework* está desarrollado completamente con PHP 5 y es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows (ZANINOTTO 2011).

Las características que se muestran a continuación constituyen el fundamento principal del uso de este *framework* en el desarrollo de la propuesta de solución:

- Facilidad en la instalación y configuración en la mayoría de plataformas más utilizadas, lo que permite a los miembros del equipo de desarrollo aumentar la eficiencia en cuanto al tiempo durante el proceso de implementación.
- Es independiente del sistema gestor de bases de datos.
- Resulta sencillo de usar en la mayoría de casos, lo que permite un mejor desenvolvimiento de los desarrolladores en el proceso de implementación de la propuesta de solución.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web, lo que contribuye a desarrollar una solución de calidad.

Conclusiones parciales

Después de realizar el estudio de la bibliografía relacionada con los sistemas para la gestión de información, enmarcados en la gestión de impresiones, se pudo identificar que las soluciones existentes

no satisfacen las necesidades que afronta el VDA. Por esta razón se propone la implementación de un sistema que sea capaz de ajustarse a estas necesidades y que se nutra de las principales ventajas que muestran algunas de las soluciones analizadas como por ejemplo la capacidad de generación de reportes, la autenticación mediante la cuenta institucional de la universidad y la característica de ser multiplataforma.

Una vez analizadas las tendencias en el desarrollo de software y de profundizar en los tipos de metodologías de desarrollo, se propuso la utilización de la metodología OpenUp. A raíz de este análisis, se definió además el uso de herramientas y tecnologías basadas en el paradigma del software libre como son Visual Paradigm en su versión 8.0, utilizando como lenguaje de modelado UML 2.0 para el proceso de diseño de artefactos, Symfony 2.3, como *framework* de desarrollo, utilizando los lenguajes de programación PHP 5.5 y Javascript 1.8.3, PostgreSQL 9.3 como SGBD, así como la herramienta PgAdmin III 1.18 para su administración. Además se decidió utilizar el entorno de desarrollo NetBeans en su versión 8.0 y como servidor web el Apache 2.2.22.

CAPÍTULO 2: "ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA"

Introducción

En el capítulo que a continuación se presenta, se exponen las tareas correspondientes al análisis y el diseño de la propuesta de solución. Se detalla el modelo de dominio, con sus diferentes clases y su diagrama de clases, el levantamiento de los requisitos tanto funcionales como no funcionales y el modelo de casos de uso del sistema con su descripción textual y diagramas. Posteriormente se muestra una fundamentación de la arquitectura y patrones de diseño utilizados, así como el modelo de datos y el de despliegue. Al finalizar el capítulo se podrá contar con una mejor comprensión acerca de la situación y características del negocio en cuestión.

2.1 Propuesta de solución

Se propone realizar un sistema informático que permita gestionar el proceso de impresiones que se lleva a cabo en el VDA de la facultad 6. Este sistema permitirá asignar recursos a áreas, así como darle seguimiento a su consumo, lo que incidirá directamente en un mejor proceso de control y organización.

2.2 Modelo de negocio

En todo proceso de informatización es necesario antes de desarrollar una solución para un escenario determinado, comprender de manera natural cómo funciona este y cuáles son sus principales características. Precisamente este el objetivo principal de un modelo de negocio.

Según Ivar Jacobson un modelo de negocio describe los procesos de negocio de una empresa en términos de casos de uso del negocio y actores del negocio, que se corresponden con los procesos del negocio y los clientes, respectivamente. Al igual que el modelo de casos de uso para un sistema software, el modelo de casos de uso del negocio presenta un sistema desde la perspectiva de su uso y esquematiza cómo proporciona valor a sus usuarios (JACOBSON *et al.* 2000).

2.2.1 Reglas del negocio

En una organización, tanto los procesos como los datos que estos manejan, están restringidos por reglas del negocio. Estas consisten en el conjunto de elementos del negocio que actúan como reguladores de la

ejecución, supervisión, control y toma de decisiones de los diferentes procesos y actividades (ABRAN and MOORE 2004). A continuación se listan las reglas del negocio:

- Las solicitudes de impresiones se realizan a los jefes de departamentos, quienes luego de aprobarlas, emiten una autorización firmada, que además especifica si el trabajo es de alta prioridad.
- La impresión de documentos en el VDA se realiza, después de haber sido presentada la autorización firmada por el jefe de departamento.
- El contenido legal de los documentos que se imprimen en el vicedecanato es responsabilidad directa del asistente del VDA, pues en su estación de trabajo es donde se lleva a cabo el proceso.
- Los documentos que se solicitan para imprimir deben estar en la extensión PDF.
- El servicio de impresiones se realiza en el horario comprendido de 2:00 pm a 4:45 pm.
- Se realiza el servicio de impresiones fuera del horario establecido solo cuando el trabajo es de alta prioridad.

2.2.2 Actores y trabajadores del negocio

Una vez identificados los procesos del negocio correspondientes a la solicitud de impresión de documentos y de asignación de recursos para este procedimiento, es posible determinar los actores y trabajadores involucrados en su realización. Los actores constituyen elementos externos que interactúan con los procesos del negocio, mientras que los trabajadores pueden ser abstracciones de personas, grupo de personas o un sistema automatizado que realiza una o varias operaciones en el negocio. A continuación se muestran los actores y trabajadores que participan en el proceso de impresiones del VDA de la facultad 6:

Tabla. 3 Actores del negocio.

Actor	Descripción
Profesor	Es el usuario que solicita la impresión de un documento determinado.
Vicedecano	Es el encargado de la asignación de los recursos para la realización del proceso de impresiones y el principal responsable de su correcto aprovechamiento.

Tabla. 4 Trabajadores del negocio.

Trabajador	Descripción
Jefe de departamento	Es el encargado de aprobar las solicitudes de impresión que realizan los profesores de su departamento y de emitir una autorización firmada.
Asistente del VDA	Es el encargado de atender las solicitudes realizadas por los profesores, una vez que estas han sido autorizadas.

2.2.3 Diagrama de Casos de Uso del Negocio (CUN)

Un diagrama de CUN describe los procesos de un negocio vinculados al campo de acción, y cómo se benefician e interactúan los socios y clientes en estos procesos (GONZÁLEZ 2013). A continuación se muestra el diagrama de CUN correspondiente a la propuesta de solución:

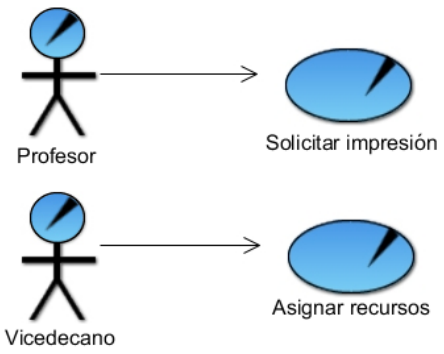

Fig. 1 Diagrama de casos de uso del negocio.

2.2.4 Descripción textual del CUN "Solicitar impresión"

Tabla. 5 Descripción textual del CUN Solicitar impresión.

Caso de Uso	Solicitar impresión
Objetivo	Imprimir un documento determinado, a partir de una solicitud y la autorización de la misma.
Actores	Profesor (inicia)

Resumen	El CUN se inicia cuando un profesor solicita a su jefe de departamento la autorización para imprimir un documento determinado y culmina cuando el asistente del vicedecano entrega al profesor el documento impreso.
Complejidad	Alta
Prioridad	Crítico
Prioridad:	Alta

Flujo de eventos "Solicitar impresión"		
Flujo básico <Solicitar impresión>		
No.	**Actor:**	**Negocio:**
1	El profesor pide la autorización para solicitar la impresión de documentos.	
2		El jefe de departamento entrega una autorización firmada.
3	El profesor realiza la solicitud de impresión al asistente del VDA.	
4		El asistente del VDA solicita la autorización firmada por el jefe de departamento.
5	El profesor entrega la autorización firmada por el jefe de departamento.	
6		El asistente del VDA recibe la autorización y comprueba que es correcta. **6.1** El asistente del VDA comprueba que los datos de la autorización están correctos. En caso de que la autorización sea incorrecta, ver el curso alterno 6.1
7		El asistente del VDA comprueba que el horario es el establecido. En caso de estar fuera de horario ver el curso alterno 7.
8		El asistente del VDA comprueba que exista

		disponibilidad técnica y recursos materiales para el servicio de impresiones. En caso de no existir alguno de estos, ver el curso alterno 8.
9		El asistente del VDA solicita el documento a imprimir.
10	El profesor entrega el documento que necesita imprimir.	
11		El asistente del VDA recibe el documento y comprueba que esté en un formato correcto. Si el formato es incorrecto, ver curso alterno 11.
12		El asistente del VDA imprime y entrega satisfactoriamente el documento impreso.
13	El profesor recibe su documento impreso.	
Cursos alternos		
6.1	El asistente del VDA rechaza la solicitud de impresión y recomienda corregir la autorización firmada por el jefe de departamento.	
7	El asistente del VDA comprueba que el nivel de prioridad de la solicitud sea alto, en caso negativo rechaza la solicitud y recomienda regresar en el horario establecido.	
8	El asistente del VDA rechaza la solicitud y explica la falta de recursos materiales o disponibilidad técnica.	
11	El asistente del VDA rechaza la solicitud y recomienda corregir el formato del documento y regresar posteriormente.	
Casos de Uso Incluidos:	No existe.	

2.2.5 Diagrama de actividades del CUN

Un diagrama de actividades muestra el orden en que se van realizando las tareas dentro de un sistema (JACOBSON *et al.* 2000). A continuación se muestra el diagrama de actividades correspondiente al CUN: Solicitar impresión.

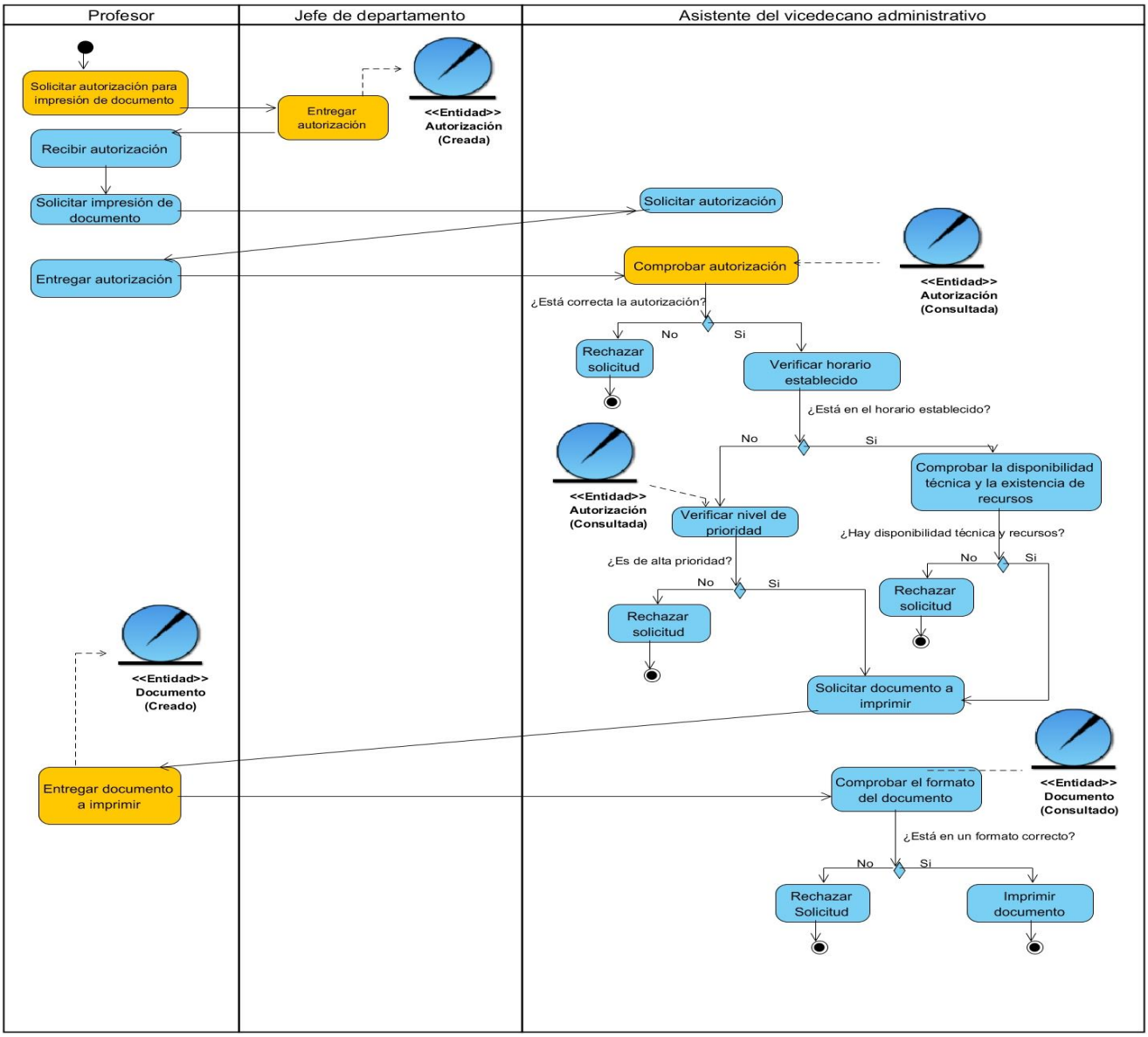

Fig. 2 Diagrama de Actividades del CUN Solicitar impresión.

2.3 Levantamiento de requisitos

Según Sommerville, durante el levantamiento u obtención de requisitos, los desarrolladores del software trabajan directamente con los clientes y los usuarios finales del sistema, para determinar el dominio de la aplicación, los servicios que debe proporcionar el sistema, el rendimiento requerido, las restricciones de

hardware, entre otros elementos, que de conjunto garantizarán la entrega de un producto final que satisfaga las necesidades planteadas por el cliente (SOMMERVILLE 2005).

Por su parte Pressman plantea que los requisitos de software son las necesidades de los clientes, los servicios que los usuarios desean que proporcione el sistema de desarrollo y las restricciones en las que debe operar (PRESSMAN 2008). En dependencia de sus características, los requisitos de software se dividen en funcionales y no funcionales, lo cual se detalla a continuación.

2.3.1 Requisitos funcionales

Los Requisitos Funcionales (RF) son los que definen las funciones que el sistema será capaz de realizar y describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. A continuación se detallan los requisitos funcionales que debe cumplir la propuesta de solución planteada.

RF 1: Autenticar usuario
RF 2: Adicionar área
RF 3: Modificar área
RF 4: Eliminar área
RF 5: Mostrar área
RF 6: Listar áreas
RF 7: Listar historial
RF 8: Borrar historial
RF 9: Adicionar solicitud
RF 10: Modificar solicitud
RF 11: Eliminar solicitud
RF 12: Mostrar solicitud
RF 13: Aprobar solicitud
RF 14: Listar solicitudes
RF 15: Filtrar solicitudes por prioridad
RF 16: Atender solicitudes aprobadas
RF 17: Enviar notificaciones al correo
RF 18: Buscar usuario
RF 19: Adicionar usuario al sistema
RF 20: Eliminar usuario del sistema
RF 21: Listar usuarios del sistema
RF 22: Adicionar recurso
RF 23: Modificar recurso
RF 24: Eliminar recurso
RF 25: Mostrar recurso
RF 26: Listar recursos
RF 27: Listar notificaciones
RF 28: Eliminar notificaciones
RF 29: Consultar notificaciones
RF 30: Configurar perfil de notificaciones
RF 31: Generar reporte de solicitudes por área
RF 32: Generar reporte de solicitudes por usuario
RF 33: Generar reporte de consumo de recursos por área

RF 34: Generar reporte de consumo de recursos por usuario.

RF 35: Generar reporte de solicitudes atendidas

RF 36: Exportar reporte en documento "*pdf*"

2.3.2 Requisitos no funcionales

Los Requisitos no Funcionales (RnF) tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento en cuanto al tiempo de respuesta, las necesidades de almacenamiento, las características definidas para las interfaces de usuario, el mantenimiento, la seguridad, la portabilidad y estándares definidos. Dentro de los requisitos no funcionales que cumple la propuesta de solución se encuentran los siguientes:

Requisitos de usabilidad:

RnF1: El sistema provee una interfaz visual amigable, sencilla e intuitiva, que garantiza la facilidad para el trabajo de los usuarios. Para ello presenta una iconografía que se ajusta en cada caso con las funciones que representan, los botones tienen descripciones acordes a las funcionalidades que realizan y en los campos de textos editables por el usuario existen etiquetas flotantes que describen la acción que se debe realizar en cada caso.

Requisitos de fiabilidad:

RnF2: La información manejada por el sistema está protegida de acceso no autorizado y divulgación. El sistema garantiza la gestión de roles y su asignación a los usuarios, lo cual permite que cada usuario tenga privilegios establecidos de acuerdo a su rol.

Requisitos de eficiencia:

RnF3: El tiempo de respuesta promedio de las peticiones realizadas al sistema no excede los 5 segundos con 200 usuarios conectados simultáneamente.

Requisitos de interfaz:

- **Interfaces de Software:**
  - **Para las computadoras cliente:**

RnF4: El sistema está desarrollado sobre tecnologías web, lo cual permite que sea multiplataforma y solamente se necesite un navegador web para acceder al mismo. Está optimizado para funcionar en el sistema operativo Nova Desktop 2013 con el navegador Mozilla Firefox en su versión 35.

- **Para los servidores del sistema:**

RnF5: Los servidores del sistema deben tener como Sistema Operativo una distribución para servidores de los sistemas basados en GNU/Linux similar o superior a la versión Ubuntu Server 12.04 LTS. Está optimizado para utilizar Ubuntu Server en su versión 14.04 LTS utilizando el servidor web Apache en su versión 2.2.22, con PHP 5 configurado y con PostgreSQL 9.3 para el manejo de la base de datos.

- **Interfaces de Hardware:**

  - **Para las computadoras cliente:**

RnF6: Las estaciones de trabajo del cliente deben contar con al menos 256 MB de memoria RAM, un procesador de 1.0 GHz como mínimo, y una tarjeta de red con velocidad de transición de 100 Mbps.

  - **Para los servidores del sistema:**

RnF7: Los servidores del sistema deben tener como mínimo 2 GB de memoria RAM, un procesador de al menos 2.0 GHz, una tarjeta de red con velocidad de transición de 100 Mbps o superior y en el caso del servidor de la base de datos, debe contar con un disco duro con más de 80 GB de capacidad de almacenamiento.

Requisitos de portabilidad:

RnF8: El sistema está desarrollado para ser utilizado en múltiples plataformas de sistemas operativos.

2.4 Modelo de casos de uso del sistema

Jacobson plantea que el modelo de casos de uso del sistema está conformado por los actores y casos de uso que interactúan en el sistema y que describe las funcionalidades que este tendrá, lo cual posibilita al cliente, a los usuarios y a los desarrolladores a llegar a un acuerdo sobre cómo utilizarlo (JACOBSON *et al.* 2000). Por su parte Larman agrega que este modelo proporciona una explicación clara y consistente de

lo que debería hacer el software, de modo que el modelo se use a lo largo del proceso de desarrollo (LARMAN 2003).

2.4.1 Casos de Uso del Sistema (CUS)

Los CUS describen cómo interactúan los actores con el software en desarrollo. Durante esta interacción, un actor genera eventos sobre un sistema, normalmente solicitando alguna operación como respuesta (LARMAN 2003). En un caso de uso se especifica una secuencia de acciones que se pueden llevar a cabo a partir de la interacción entre el sistema y los actores.

2.4.2 Actores del sistema

Los actores constituyen cada uno de los roles que agrupan a los diversos tipos de usuarios que interactúan con los diferentes CUS. A partir del análisis de las características que debe tener la solución propuesta se definen para la misma los siguientes actores:

Tabla. 6 Actores del sistema.

Actor:	Descripción:
Usuario	Accede al sistema y una vez autenticado puede gestionar las solicitudes de servicios de impresión.
Jefe de área	Se encarga de aprobar las solicitudes creadas por los miembros de su área y puede obtener reportes del comportamiento del consumo de recursos de su área.
Vicedecano administrativo	Puede monitorear todo el proceso de gestión de impresiones y acceder a reportes del comportamiento del consumo de recursos, tanto de un área en específico como a nivel general.
Asistente del vicedecano	Se encarga de imprimir las solicitudes que con antelación han sido aprobadas por cada jefe de área. En casos puntuales, también puede imprimir solicitudes que aún no hayan sido aprobadas.
Administrador	Es el responsable de todo lo relacionado con la gestión de roles y privilegios del sistema.

2.4.3 Diagrama de CUS

Los diagramas de CUS describen parte del modelo de casos de uso y muestran un conjunto de casos de uso y actores con una asociación entre cada par de estos que interactúan en el sistema (JACOBSON *et al.* 2000). A esto se puede agregar que constituye una excelente representación del contexto del sistema, ya que sirve como herramienta de comunicación para visualizar, especificar, resumir y documentar el comportamiento de este y sus actores.

A continuación se presenta el diagrama de CUS correspondiente a la solución propuesta, en el cual se representan los 14 CU que agrupan a los 36 RF identificados.

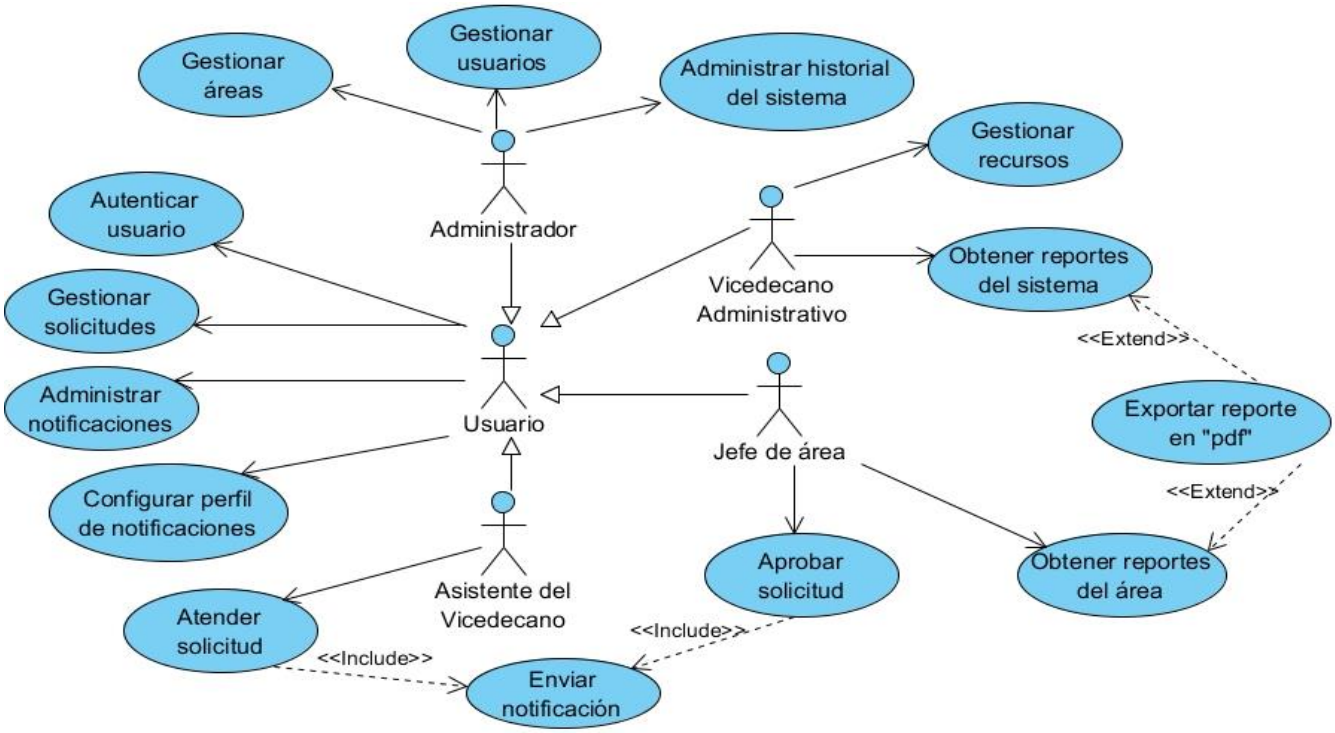

Fig. 3 Diagrama de casos de uso del sistema.

- **Descripción textual del CUS Gestionar solicitudes.**

La descripción textual o especificación de los CU tiene como objetivo describir en detalle el flujo normal de eventos así como los flujos alternos, de una manera precisa y fácilmente comprensible. Además se

incluyen las precondiciones y poscondiciones y se define cómo comienza, termina e interactúa cada CU con sus actores.

A continuación se presenta la descripción textual del CUS Gestionar solicitudes. Para profundizar en las descripciones textuales del resto de los CU ver el artefacto “0114_Especificación de CUS”.

Tabla. 7 Descripción textual del CUS Gestionar solicitudes.

Caso de Uso	Gestionar solicitudes	
Objetivo	Este CU se desarrolla con el objetivo de realizar las solicitudes de impresión del usuario, para que posteriormente sean aprobadas por el Jefe del área a la que pertenece el mismo.	
Actores	Usuario (inicia)	
Resumen:	El CU se inicia cuando el usuario selecciona la opción de crear una nueva solicitud y culmina con la realización de una de las siguientes operaciones sobre una solicitud: Adicionar, Modificar, Eliminar o Mostrar.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	El usuario debe estar autenticado en el sistema y debe tener privilegios para realizar las acciones con las que puede culminar el CU. En caso de que se requiera Mostrar, Eliminar o Modificar alguna solicitud, que anteriormente haya sido creada, se debe seleccionar de la lista de solicitudes y posteriormente seleccionar la operación.	
Poscondiciones	Al finalizar el CU la solicitud se crea, se modifica, se elimina o se muestra, en dependencia de la opción seleccionada por el usuario.	
Flujo de eventos		
Flujo básico <Gestionar solicitudes>		
No.	**Actor:**	**Sistema:**
1	Selecciona la pestaña “Gestionar	

	solicitudes" y escoge la operación a realizar.	
2		Despliega dos nuevas pestañas que muestran las opciones "Adicionar solicitud" y "Mis solicitudes".
3	Selecciona la acción de adicionar una nueva solicitud o a partir de la lista escoge modificar, eliminar o mostrar una solicitud.	
4		Si el usuario escoge la opción "Adicionar solicitud", ver la sección "Adicionar solicitud"; en caso de escoger la opción "Mis solicitudes", ver la sección "Mis solicitudes".
Sección 1: "Adicionar solicitud"		
Flujo básico <Adicionar solicitud>		
No.	**Actor:**	**Sistema:**
1		Muestra una interfaz para introducir el nombre, la cantidad de copias y el tipo de descuento de recurso (en caso de que el usuario tenga recursos asignados) de la nueva solicitud.
2	Introduce los datos requeridos y selecciona las opciones "Adicionar" o "Aplicar".	
3		Comprueba que no existan campos vacíos. Si existen campos vacíos, ver el flujo alterno 3. **3.1** El sistema comprueba que los datos introducidos en los campos sean correctos. Si existen campos con datos no válidos, ver el flujo alterno 3.1.
4		Adiciona una nueva solicitud, si se realiza utilizando los recursos de un área, le asigna el estado "En espera" y envía una notificación al Jefe del área a la que pertenece el usuario. Esta acción da paso a la ejecución del CU Aprobar

		Solicitud. Si se realiza usando recursos personales pasa directamente al estado "Aprobada", acción que da paso al CU Atender Solicitud.
Flujos alternos al paso 3		
No.	**Actor:**	**Sistema:**
3		Informa al usuario que debe rellenar los campos vacíos mediante un mensaje de error.
3.1		Informa al usuario sobre la existencia de campos con datos no válidos mostrando el mensaje de error: "Ha introducido datos inválidos"
Sección 2: "Mis solicitudes"		
Flujo básico <Mis solicitudes>		
No.	**Actor:**	**Sistema:**
1		Muestra una interfaz con la lista de las solicitudes que el usuario previamente ha insertado en el sistema, con las opciones para "Mostrar", "Actualizar" y "Eliminar".
2	Escoge una de las opciones que muestra el sistema.	
3		Si el usuario escoge la opción "Mostrar" ver la sección "Mostrar solicitud", si el usuario escoge la opción "Actualizar" ver la sección "Actualizar solicitud", si el usuario escoge la opción "Eliminar" ver la sección "Eliminar solicitud".
Sección 3: "Mostrar solicitud"		
Flujo básico <Mostrar solicitud>		
No.	**Actor:**	**Sistema:**
1		Muestra una interfaz con los datos correspondientes a la solicitud seleccionada.
Sección 4: "Actualizar solicitud"		
Flujo básico <Actualizar solicitud>		
No.	**Actor:**	**Sistema:**

1		Muestra una interfaz para introducir los datos a actualizar en la solicitud.
2	Introduce los datos requeridos y selecciona la opción “Modificar”	
3		Comprueba que no existan campos vacíos. Si existen campos vacíos, ver el flujo alterno 3. **3.1** El sistema comprueba que los datos introducidos en los campos sean correctos. Si existen campos con datos no válidos, ver el flujo alterno 3.1
4		Actualiza la solicitud seleccionada con los nuevos datos introducidos.
Flujos alternos al paso 3		
No.	**Actor:**	**Sistema:**
3		Informa al usuario que debe rellenar los campos vacíos mediante un mensaje de error.
3.1		Informa al usuario sobre la existencia de campos con datos no válidos mostrando el mensaje de error: “Ha introducido datos inválidos”
Sección 5: “Eliminar solicitud”		
Flujo básico <Eliminar solicitud>		
No.	**Actor:**	**Sistema:**
1		Lanza un mensaje para que el usuario confirme que desea eliminar la solicitud seleccionada.
2	Selecciona la opción “Eliminar”	
3		Elimina la solicitud seleccionada, así como los documentos asociados a esta.

Relaciones	CU Incluidos	No procede
	CU Extendidos	No procede
Requisitos no funcionales	No procede	
Asuntos pendientes	No procede	

2.4.4 Patrones de casos de uso utilizados

Durante la construcción del diagrama de CUS se utilizaron los patrones de CU que se detallan a continuación:

- **CRUD (Creating, Reading, Updating, Deleting) Completo**:

Este patrón consta de un CU, llamado Información CRUD o Gestionar información, modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación. Suele ser utilizado cuando todos los flujos contribuyen al mismo valor del negocio, y estos a su vez son cortos y simples. Este patrón se pone de manifiesto en los CU Gestionar áreas, Gestionar recursos, Gestionar usuarios y Gestionar solicitudes.

- **CRUD Parcial**:

Se considera una versión del CRUD Completo y su principal diferencia con este radica en que una o varias de las operaciones (creación, lectura, actualización y eliminación) no están presentes en el CU. Su aplicación se evidencia en los CU Administrar historial y Administrar notificaciones.

- **Generalización / Especialización entre actores**:

Permite agrupar varios actores que comparten similares roles con respecto a un CU determinado. En la generalización de actores existe una clase padre de actor y una clase hija de actor que es una especialización de la clase padre. En estos casos la hija hereda todas las características de la clase padre y además puede agregar nuevas características. Está presente en el DCUS cuando el administrador, el asistente del vicedecano, el vicedecano administrativo y el jefe de área se comportan como un usuario, que se relaciona con Autenticar usuario, Gestionar solicitudes, Administrar notificaciones y Configurar perfil de notificaciones.

- **Extensión Concreta o Inclusión**

Este patrón incluye la extensión concreta, en la que se presenta un CU que extiende o añade, con el cual se describe dónde y bajo qué condiciones extiende el comportamiento de algún CU base

(LARMAN 2003). La extensión concreta está presente por ejemplo en los CU relacionados con la obtención de reportes, los cuales extienden a “Exportar reporte a pdf”. También en este patrón recoge la inclusión concreta, la cual es una relación desde un CU base a uno de inclusión, que especifica cómo el comportamiento definido para el de inclusión se inserta explícitamente dentro del comportamiento definido para el CU base. Esta relación se evidencia en “Atender solicitud”, que incluye a “Enviar notificación”.

2.5 Elementos fundamentales del diseño y arquitectura del sistema.

El diseño del sistema tiene como propósitos fundamentales adquirir una profunda comprensión de los aspectos relacionados con los requisitos funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos y tecnologías utilizadas. Permite crear un punto de partida para la actividades de la implementación, descomponiéndolas en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo (JACOBSON *et al.* 2000).

2.5.1 Estilo Arquitectónico

Un estilo arquitectónico es una lista de tipos de componentes que describen los patrones o las interacciones a través de ellos. Un estilo afecta a toda la arquitectura de software y puede combinarse en la propuesta de solución. Los estilos ayudan a un tratamiento estructural que concierne más bien a la teoría, la investigación académica y la arquitectura en el nivel de abstracción más elevado, expresando la arquitectura en un sentido más formal y teórico (ALMEIRA 2007).

Para el desarrollo de la propuesta de solución se decide el uso del estilo de Llamada y Retorno, en el cual el sistema se constituye de un programa principal que lo controla y varios subprogramas que se comunican con él. Según Pressman el empleo de este estilo posibilita además la comunicación, la coordinación y cooperación entre los componentes y las restricciones que definen cómo se integran para conformar el sistema, así como los modelos semánticos que facilitan al diseñador el entendimiento de todas las partes del sistema, evitando que las variaciones realizadas a funcionalidades o componentes específicos afecten el funcionamiento general (PRESSMAN 2008).

2.5.2 Arquitectura y Patrones de Diseño

Según Sommerville, el diseño arquitectónico o arquitectura del software es la primera etapa en el proceso de diseño y representa un enlace crítico entre los procesos de ingeniería de diseño y de requerimientos. Está relacionado con el establecimiento de un marco estructural básico que identifica los principales componentes de un sistema y las comunicaciones entre estos componentes (SOMMERVILLE 2005).

Patrón arquitectónico Modelo Vista Controlador (MVC)

Un patrón arquitectónico brinda la descripción de un problema particular y recurrente de diseño, que aparece en contextos de diseño específico, y presenta un esquema genérico demostrado con éxito para su solución.
Para el desarrollo de la propuesta de solución se define el patrón arquitectónico MVC, el cual se basa en las ideas de reutilización de código y la separación de conceptos, buscando facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. Su principal característica radica en que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos, el modelo, la vista y el controlador, los cuales se detallan a continuación:

El modelo: es la capa donde se trabaja con los datos, por tanto contendrá mecanismos para acceder a la información y también para actualizar su estado. En la solución propuesta esta capa agrupa los paquetes “Entity” y “Model”. El primero incluye dos paquetes: “Entity”, que recoge las entidades del sistema que se crean a partir de las tablas de la base de datos, y “Repository” que contiene entidades que se encargan del manejo y selección de datos. Por su parte, en el paquete “Model” se incluyen las clases del dominio y la lógica de datos.

Las vistas: se encargan de presentar la información del sistema al usuario y generar los eventos de la interacción con éste. Capturan eventos del usuario y se los envía al sistema a través del controlador. Posteriormente reciben una respuesta del controlador y muestran la información al usuario. En la propuesta de solución esta capa contiene los paquetes “View”, “JQuery” y “CSS”. El primero recoge las páginas y plantillas que conforman al sistema, así como los archivos twig. El paquete “JQuery” contiene los archivos javascript y el “CSS” las hojas de estilo.

El controlador: contiene el código necesario para responder a las acciones que se solicitan en la aplicación, sirve de enlace entre las vistas y el modelo, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de la aplicación. En la solución propuesta esta capa contiene los controladores del sistema, quienes se encargan de crear y devolver una respuesta, una vez que han recibido una petición. Para esto se utilizan clases que son destinadas a la lógica de control y otras correspondientes a los formularios, las cuales se agrupan en los paquetes "Controller" y "Form" respectivamente.

En la figura 4 se muestra el diagrama de paquetes del diseño correspondiente al CU Gestionar solicitudes donde se evidencian los elementos que conforman al modelo, las vistas y el controlador, que anteriormente se explican:

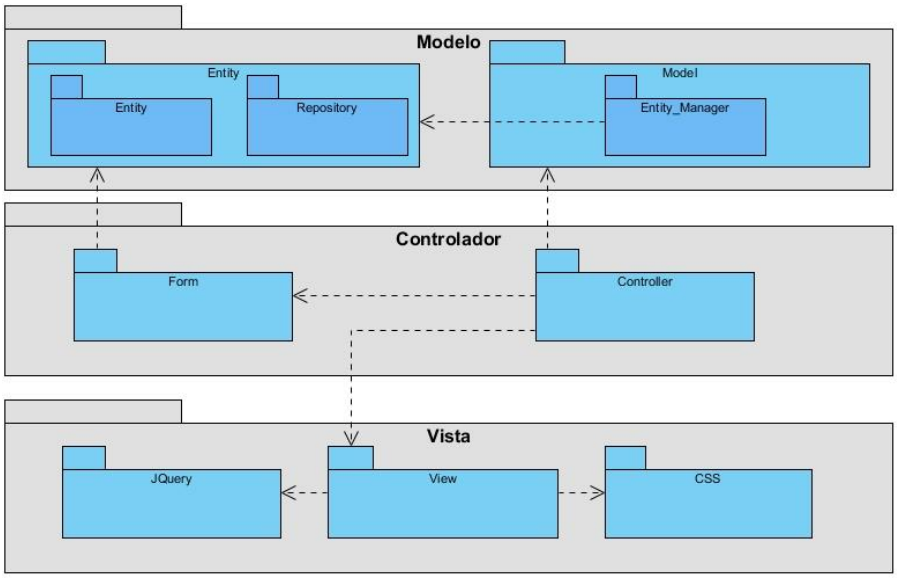

Fig. 4 Diagrama de paquetes correspondiente al CU Gestionar solicitudes.

Patrones de diseño

Los patrones de diseño constituyen una descripción de un problema y la solución, a la que se da un nombre y que se puede aplicar a nuevos contextos; idealmente, proporcionan consejos sobre el modo de aplicarlos en varias circunstancias, y consideran los puntos fuertes y compromisos. Muchos patrones proporcionan guías sobre el modo en el que deberían asignarse las responsabilidades a los objetos, dada una categoría específica del problema (LARMAN 2003).

Patrones de Principios Generales para Asignar Responsabilidades (GRASP)

Los patrones GRASP describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. A continuación se explica cómo se evidencia el uso de estos patrones en la propuesta de solución.

- **Experto:** propone asignar las responsabilidades a las clases de acuerdo a la información que contienen las mismas cumpliendo así un principio básico e intuitivo de la programación orientada a objetos. Se utiliza con frecuencia en la asignación de responsabilidades; es un principio de guía básico que se utiliza continuamente en el diseño de objetos (LARMAN 2003). Su utilización se pone de manifiesto ya que Symfony utiliza el objeto relacional de mapeo (*ORM* por sus siglas en inglés) Doctrine, para la capa del modelo. Doctrine se encarga de crear una clase experta por cada tabla de la base de datos del modelo, como se puede apreciar en la figura 5 en la clase "Solicitudes", esto permite que se pueda manejar su información como un objeto de tipo la entidad mapeada.

- **Controlador:** sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, manejando los eventos de entrada de dicha interfaz. Es un objeto que no pertenece a la interfaz de usuario, responsable de recibir o manejar un evento del sistema, sino que define el método para la operación del mismo (LARMAN 2003). Este patrón está presente en la propuesta de solución pues se crearon varios controladores que se encargan de manejar eventos entre la vista y el modelo, como por ejemplo en la figura 5 el controlador "SolicitudController".

- **Creador:** guía la asignación de responsabilidades relacionadas con la creación de objetos, una tarea muy común. La intención básica del patrón Creador es encontrar un creador que necesite conectarse al objeto creado en alguna situación. Eligiéndolo como el creador se favorece el bajo acoplamiento (LARMAN 2003). El uso de este patrón se evidencia en algunas clases en la capa del modelo, que se encargan de crear instancias de las clases que proveen la información necesaria para su propio manejo, como por ejemplo la clase "SolicitudController" en la figura 5.

- **Bajo acoplamiento:** propone la asignación de responsabilidades de manera tal que la dependencia entre una clase y otra sea la menor posible, de tal forma que se potencie la reutilización y se mitiguen los efectos que puedan producir en una, la realización de cambios en la

otra. Para esto soporta el diseño de clases que son más independientes, lo que reduce el impacto del cambio (LARMAN 2003). Se pone de manifiesto en la solución propuesta pues entre las clases del controlador y la vista o el controlador y el modelo, existe una baja interdependencia, lo que se traduce en la posibilidad de efectuar cambios en estas sin que ocurran grandes afectaciones al resto del sistema.

- **Alta cohesión:** se basa en que los elementos de un componente colaboran para producir algún comportamiento bien definido, como una clase que tiene una responsabilidad moderada en un área funcional y colabora con otras clases para llevar a cabo las tareas (LARMAN 2003). Se manifiesta el uso de este patrón en la propuesta de solución pues por ejemplo, como se puede apreciar en la figura 5, las clases “Solicitud” y “SolicitudController” muestran responsabilidades relacionadas coherentemente, que se complementan entre sí, lo cual garantiza que exista además un bajo acoplamiento que favorece el equilibrio y un diseño en el cual los objetos sean capaces de interactuar.

Patrones de la Banda de los Cuatro (GoF)

Los patrones GoF (por sus siglas en inglés de *The Gang of Four*), se utilizan para diseñar objetos y solucionar problemas de creación de instancias, ya que ayudan a encapsular y abstraer dicha creación. A continuación se explica cómo se manifiestan estos patrones en la propuesta de solución:

- **Decorador:** es un patrón de tipo estructural que permite añadir dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender funcionalidades (LARMAN 2003). Este patrón está presente en el diseño de la propuesta de solución, pues en la misma se utilizan plantillas twig para el desarrollo de las vistas. Estas plantillas heredan de la plantilla “base.html.twig”, lo cual permite declarar regiones o bloques editables dentro de los cuales se realizan las modificaciones particulares de cada página, manteniendo una apariencia homogénea entre todas.

- **Singleton:** se manifiesta al garantizarse que una clase solo tenga una única instancia, la cual proporcione un único punto de acceso local a esta (LARMAN 2003). Este patrón se evidencia por ejemplo en la clase “Manager” encargada de acceder a las funcionalidades de Doctrine, la cual al

constituir una única instancia es accedida directamente sin que exista la posibilidad de crear nuevos objetos de su tipo.

- **Agente remoto:** este patrón se pone de manifiesto cuando el sistema requiere comunicarse con un servicio externo y no se desea o no es posible acceder a este directamente (LARMAN 2003). La utilización de este patrón en el sistema se evidencia en la autenticación mediante el servicio "Ldap" disponible para la UCI, el cual provee los datos de los usuarios necesarios para desarrollar este proceso de manera segura. Para evitar acceder directamente a estos datos se creó la clase "Ldap.php", que es utilizada como mediadora por la clase "LoginController", para obtener los datos, comprobarlos y proceder a la autenticación.

2.5.3 Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema (JACOBSON *et al.* 2000).

2.5.4 Diagrama de clases del diseño

Un Diagrama de Clases de Diseño (DCD) representa los detalles de las especificaciones de las clases e interfaces software en una aplicación. A diferencia de las clases conceptuales del modelo del dominio, las clases de diseño de los DCD muestran las definiciones de las clases del software en lugar de los conceptos del mundo real (LARMAN 2003).

En la figura 5 se muestra el DCD correspondiente al CU Gestionar solicitudes, donde se puede observar la relación directa entre las 3 capas de la arquitectura MVC. En el modelo se encuentran las clases "Solicitud", "Documento" y "Usuario" que acceden a los datos y se relacionan con el controlador "SolicitudController", que es el encargado de comunicarse con el modelo y recibir los objetos de tipo *Request* del controlador frontal del sistema ("SP_App"). Este último, a su vez, obtiene mediante procedimientos internos del *framework* las solicitudes de las vistas en la capa superior, a continuación ordena la ejecución de la acción correspondiente en el controlador, para finalmente devolver el objeto *Response* que es interpretado por el navegador para mostrar la información al usuario mediante las vistas.

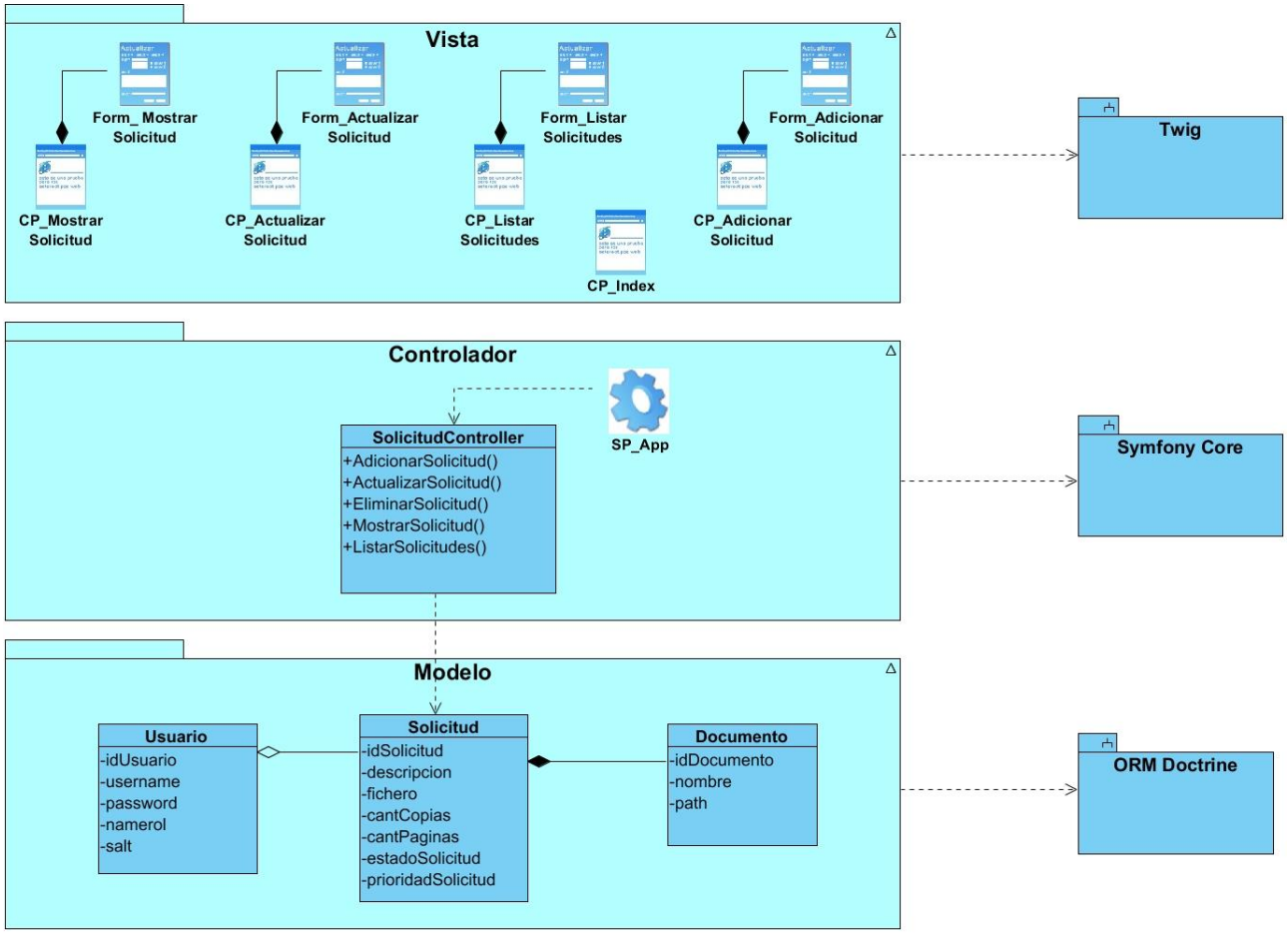

Fig. 5 Diagrama de clases del diseño correspondiente al CU Gestionar solicitudes.

2.5.5 Diagrama entidad – relación

El Diagrama Entidad Relación (E-R) proporciona una herramienta para representar información del mundo real a nivel conceptual. Permite describir las entidades involucradas en una base de datos, así como las relaciones y restricciones de ellas (GAONA 2012). En la figura 6 se muestra el diagrama E-R correspondiente al modelo de datos de la propuesta de solución, donde se agrupan las entidades que lo conforman y las relaciones entre estas. Se puede apreciar que la entidad “Usuario” puede tener asociado un perfil o un historial y uno o varios elementos de tipo mensaje, solicitud o recurso de usuario. Por su parte la entidad “Area” puede tener asociado uno o varios elementos de tipo usuario y recurso de área, así como la entidad “Solicitud” puede asociarse con uno o varios documentos.

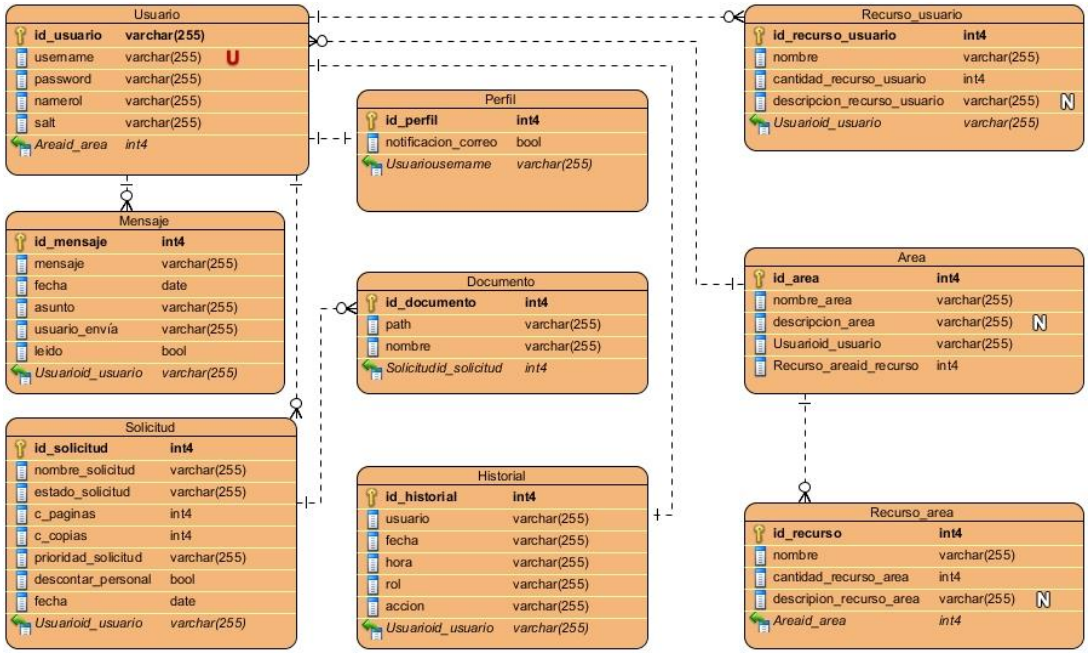

Fig. 6 Diagrama E-R de la propuesta de solución.

Conclusiones parciales

En el presente capítulo se realizó el análisis del negocio en cuestión a partir del cual se pudo identificar con el cliente 36 RF y 8 RnF. Se especificaron además los CU, que además fueron relacionados mediante el diagrama de CUS y descritos formalmente.

Una vez realizado el diseño de la propuesta de solución, se obtuvo el diagrama E-R del modelo de datos y los diagramas de clases del diseño, resaltando los patrones de diseños que se ponen de manifiesto en estos últimos.

CAPÍTULO 3: "IMPLEMENTACIÓN Y PRUEBAS"

Introducción

En el presente capítulo se abordan los principales aspectos referentes al flujo de implementación y pruebas. Se describe la distribución física del sistema propuesto a través del modelo de despliegue y se representa el sistema dividido en componentes y dependencias entre estos mediante el diagrama de componentes. Además se explican los estilos de programación y estándares de codificación empleados y finalmente se define el modelo de pruebas, reflejando a la vez el resultado de la ejecución de las mismas.

3.1 Modelo de implementación

El modelo de implementación describe cómo los elementos de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente y ejecutables. Además describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en los lenguajes de programación utilizados, y cómo dependen los componentes unos de otros (JACOBSON *et al.* 2000).

3.1.1 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos y sus relaciones en el entorno. Los componentes representan los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas, y las relaciones de dependencia se utilizan para indicar que un componente utiliza los servicios ofrecidos por otro componente. Además agrupan varias partes de un sistema modular, desplegables y reemplazables, que encapsulan la implementación y expone un conjunto de interfaces, como por ejemplo, código fuente, binario o ejecutable (LARMAN 2003).

En la figura 7 se muestra el diagrama de componentes correspondiente al CU Gestionar solicitudes. En el mismo se representan las principales dependencias entre los elementos fundamentales que conforman la estructuración del sistema ubicados en el paquete ***"Sistema"*** y la estructuración del *framework,* representados en el paquete ***"Symfony 2"***. Dentro de los elementos que componen al sistema se puede apreciar el paquete ***"View"***, que agrupa las vistas correspondientes a cada una de las diferentes acciones que se ejecutan en el caso de uso, que constituyen extensiones "html.twig" y que heredan del componente

"base.html.twig"; en el paquete ***"Model"*** se encuentra el "SolicitudEntity" encargado de manejar los elementos referentes al negocio; en el paquete ***"Controller"*** se ubica el "SolicitudController", que se encarga de manejar la lógica del control del sistema y el "SolicitudForm" que contiene los formularios utilizados por los controladores; y en el paquete ***"Resources"*** se ubican los recursos "JQueryUI", "DataTable.js" y "Style.css" que agrupan los elementos encargados de dar estilo y el dinamismo al sistema.

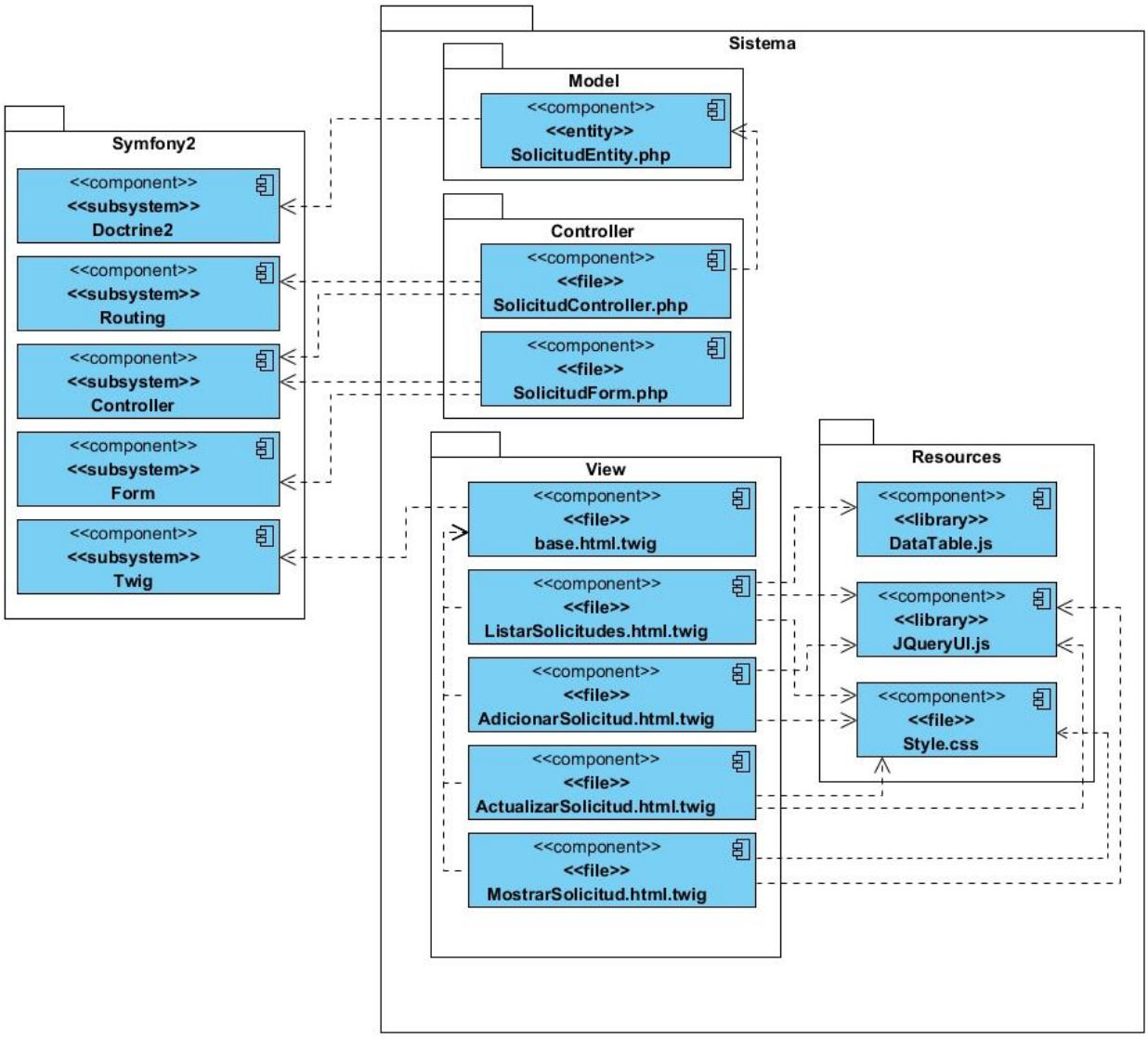

Fig. 7 Diagrama de componentes CU Gestionar solicitudes.

3.2 Modelo de despliegue

Un modelo de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final. Según Jacobson se le puede considerar como una distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo, además agrega que el modelo de despliegue se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño (JACOBSON *et al.* 2000).

En la figura 8 se muestra el diagrama correspondiente al modelo de despliegue de la propuesta de solución, donde se representan físicamente el servidor de la aplicación, el de la base de datos, el servicio web Ldap disponible para la UCI y la estación cliente, que en el caso de la del asistente del vicedecano tendrá conectada una impresora, para atender las solicitudes. Además en cada relación se representan los diferentes protocolos de comunicación.

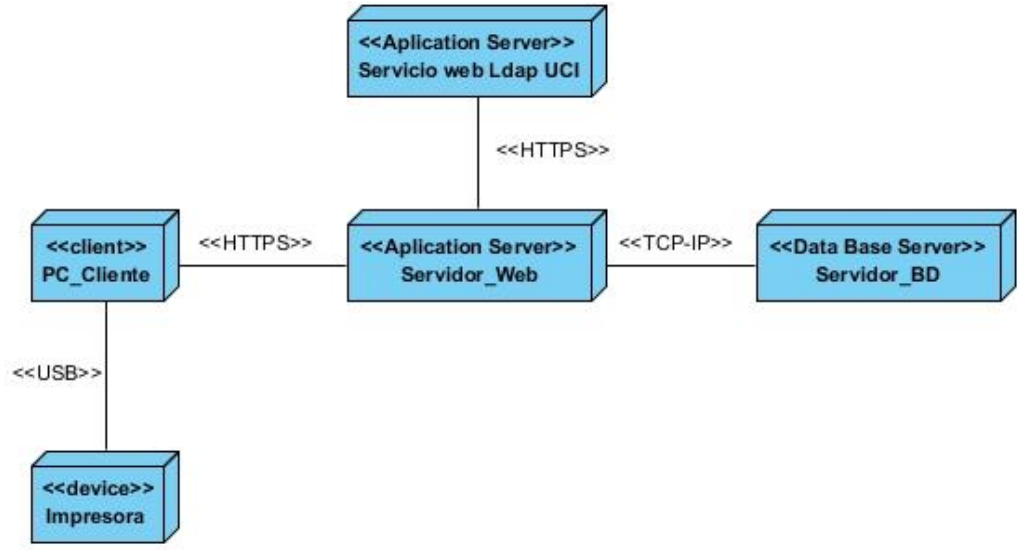

Fig. 8 Diagrama del modelo de Despliegue.

3.3 Código fuente

El código fuente de un sistema informático constituye el conjunto de líneas de texto que contienen las instrucciones que deben ser procesadas mediante compiladores, ensambladores o intérpretes hacia el lenguaje código de máquina para su posterior ejecución. En diversas ocasiones la calidad de un software

depende en gran medida de las buenas prácticas que se manifiesten en su proceso de implementación. A continuación se muestran los principales elementos referentes a la implementación de la propuesta de solución.

3.3.1 Estándares y estilos de codificación

El propósito fundamental de los estándares de codificación es que el sistema tenga una arquitectura y un estilo consistente, independiente del autor, con lo cual resulte fácil de entender y por supuesto fácil de mantener. Partiendo de que la propuesta de solución se desarrolla utilizando el *framework* Symfony 2.3, se define el uso de los estándares PSR-1 *(PHP Standards Recommendation 1)* y PSR-2 *(PHP Standards Recommendation 2)* propuestos por esta tecnología y que de conjunto constituyen unos de los estándares más difundidos dentro del lenguaje de programación PHP (POTENCIER *et al.* 2014). Estos estándares plantean fundamentalmente los siguientes elementos:

- Se deben utilizar solamente las etiquetas <?php y <?=.
- Se debe emplear solamente la codificación UTF-8 para el código PHP.
- El código debe usar 4 espacios como indentación, no tabuladores.
- Debe haber una línea en blanco después de la declaración del *"namespace"* y otra después del bloque de declaraciones *"use"*.
- Las llaves de apertura de las estructuras de control deben estar en la misma línea, y las de cierre deben ir en la línea siguiente al cuerpo.
- Los paréntesis de apertura en las estructuras de control no deben tener un espacio después de ellos, y los paréntesis de cierre no deben tener un espacio antes de ellos.

En el caso de las convenciones de nomenclatura que hacen uso de las mayúsculas y minúsculas en sus identificadores se utilizan cuatro estilos:

- El estilo **Pascal** (PascalCase) que plantea que la primera letra del identificador y la primera letra de las siguientes palabras concatenadas están en mayúsculas, se utiliza para identificar las clases.
- El estilo **Camel Case** (camelCase) que define que la primera letra del identificador debe estar en minúscula y la primera letra de las siguientes palabras concatenadas en mayúscula, se utiliza para identificar los métodos y los parámetros que pudiesen contener estos.
- El estilo **Mayúsculas** (ALL_CAPS o UPPER_CASE) que plantea que todas las letras deben ir en mayúsculas se utiliza para identificar las constantes.

- El estilo **Minúsculas** (small_caps o lower_case) que precisa que todas las letras deben ir en minúsculas se utiliza para identificar las variables.

A continuación se muestra un fragmento del código fuente de la propuesta de solución donde se evidencia el uso de estándares y estilos de codificación anteriormente planteados. En el primer caso se muestra un fragmento del CU “Enviar notificación” y en el segundo caso un fragmento del CU “Administrar historial”

```
private function enviarCorreo($sujeto, $de, $usuario, $cuerpo)
{
    $persona = $this->get('println.ldap')->damePersona($usuario);
    if ($persona->Cargo->NombreCargo == 'Estudiante') {
        $correo = $usuario . '@estudiantes.uci.cu';
    } else {
        $correo = $usuario . '@uci.cu';
    }
    $mensaje = \Swift_Message::newInstance()
        ->setSubject($sujeto)
        ->setFrom($de)
        ->setTo($correo)
        ->setBody($cuerpo);
    $this->get('mailer')->send($mensaje);
}
```

Fig. 9 Fragmento del código fuente CU: Enviar notificación.

```
    //Guardar historial
    $historial = new Historial();
    $this->get('session');
    $nombre_area = $em->getRepository('printlnBundle:Area')-
>findById($entity->getAreaidArea());
    $historial->setUsuario($session->get('usuario'));
    $historial->setFecha(date("d-m-Y"));
    $historial->setHora(date("H:i:s"));
    $historial->setRol($session->get('rol'));
    $historial->setAccion("Se ha eliminado el usuario: " . $entity-
>getUsername() . " que pertenecía al área: " . $nombre_area[0]-
>getNombreArea() . '.');
    $em->persist($historial);
    $em->flush();
```

Fig. 10 Fragmento del código fuente CU: Administrar historial.

3.4 Modelo de pruebas

El modelo de pruebas describe principalmente cómo se prueban los componentes ejecutables en el modelo de implementación con pruebas de integración y de sistema, así como otros aspectos específicos del sistema (JACOBSON *et al.* 2000). Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente (IEEE 1995).

Según Pressman los objetivos fundamentales del proceso de prueba están dados por las siguientes afirmaciones:

- La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubre un error no detectado hasta entonces (PRESSMAN 2008).

3.4.1 Tipos de pruebas

Existen varios tipos de pruebas, algunas recomendadas específicamente para aplicaciones web, como las que se utilizan en esta investigación y que a continuación se explican:

Pruebas de función o a nivel de componentes, que ejercitan el contenido y las unidades funcionales dentro de la aplicación y se enfocan sobre un conjunto de pruebas que intentan descubrir errores en la misma.

Pruebas de desempeño o de rendimiento, que se aplican para descubrir problemas debido a la falta de recursos en el lado del servidor, ancho de banda de red inapropiado, capacidades inadecuadas de bases de datos, defectuosas o débiles capacidades del sistema operativo, funcionalidades mal diseñadas y otros conflictos de hardware o software que pueden conducir a un pobre desempeño cliente - servidor (PRESSMAN 2008).

Pruebas de aceptación, que representan aquella fase del ciclo de vida de desarrollo de software en el que el equipo de desarrollo y el área usuaria de un sistema de información tienen que garantizar que el sistema desarrollado se corresponde con los requerimientos definidos (GONZÁLEZ, JESUS PONCE et al. 2014). En la presente investigación se utilizaron estas pruebas específicamente las de tipo Alfa, que son las que realiza el usuario final, una vez recibido el producto terminado y su documentación, de conjunto

con los desarrolladores del sistema. Para ello se le entregó al vicedecano de administración el software terminado, junto a una guía para el desarrollo de estas pruebas. Este proceso se realizó en el VDA de la facultad 6, en presencia de los desarrolladores y los especialistas del área, que se encargaron de comprobar todas las funcionalidades y de informar de las deficiencias y errores que detectaron.

3.4.2 Métodos y técnicas de prueba

Para desarrollar las pruebas de función o a nivel de componentes se aplica el método de prueba de **Caja Negra**, también denominado Pruebas de Comportamiento. Este método permite obtener un conjunto de condiciones de entrada que ejerciten por completo los requisitos funcionales de un programa, ignorando la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

Las pruebas de Caja Negra según Pressman buscan encontrar errores en cinco categorías:

- Funciones incorrectas o ausentes
- Errores de interfaz
- Errores en estructuras de datos o en accesos a bases de datos externas
- Errores de rendimiento
- Errores de inicialización y terminación

Dentro del método de Caja Negra se decide el uso de la técnica de Partición de equivalencia, que es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software. Además esta técnica se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así el número de clases de pruebas a desarrollar. Para su implementación se divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software a partir de las cuales pueden derivarse casos de prueba (PRESSMAN 2008).

Para desarrollar las pruebas de desempeño o de rendimiento se pueden aplicar pruebas de carga y de tensión. En el caso de la presente investigación debido a que la propuesta de solución durante su despliegue no estará expuesta a grandes niveles de tensión ni concurrencia significante de usuarios, se decide realizar solamente **pruebas de carga**, que tienen como objetivo determinar cómo la aplicación en su ambiente del lado del servidor responderá ante varias condiciones de carga, a partir de condiciones de pruebas definidas por las permutaciones entre variables que dependen del número de usuarios

concurrentes, el número de transacciones en línea por usuario por unidad de tiempo y la carga de datos procesada por el servidor por transacción (PRESSMAN 2008).

3.4.3 Diseño de Casos de Prueba (DCP)

Un caso de prueba constituye el conjunto de entradas, condiciones de ejecución y resultados esperados que se desarrollada para validar una funcionalidad específica en la aplicación bajo prueba. La calidad de las pruebas es proporcional al número de casos de prueba, debido a que cada caso de prueba refleja diferentes escenarios, condiciones, o flujo de trabajo (LEWIS 2005). A continuación se muestra el DCP del CU Gestionar solicitudes. Para profundizar en el resto de los DCP realizados, se deben consultar los artefactos del modelo de pruebas del expediente de proyecto de la propuesta de solución.

DCP del caso de uso: Gestionar solicitudes

- **Descripción general**

El presente CU se inicia cuando el usuario despliega la opción Gestionar solicitudes, selecciona las operaciones de adicionar una nueva solicitud o listar las solicitudes existentes y culmina con la realización de una de las acciones *Adicionar*, *Editar*, *Mostrar o Eliminar* alguna solicitud.

- **Condiciones de ejecución**

El usuario debe estar autenticado en el sistema, así como contar con los permisos para realizar alguna de las operaciones antes mencionadas. En el caso de que se pretenda *Actualizar*, *Mostrar o Eliminar* alguna solicitud, esta debe haber sido adicionada anteriormente al sistema. En el caso de que se pretenda actualizar una solicitud, el usuario solo podrá realizar la acción si la misma no ha sido aprobada, en otro caso solo podrá eliminarla y consultar sus detalles.

- **Secciones a probar en el CU: Gestionar solicitudes**

Tabla. 8 DCP Gestionar solicitudes.

Nombre de la	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central

sección			
SC 1: "Adicionar Solicitud".	**EC 1.1:** El usuario inserta los datos requeridos de manera correcta.	El sistema adiciona una nueva solicitud y envía una notificación al usuario que confirma que su solicitud se creó correctamente.	1. Seleccionar la pestaña "Gestionar solicitudes" en el menú de la izquierda. 2. Seleccionar dentro de esta pestaña la opción "Adicionar solicitud". 3. Insertar los datos en los campos, según se corresponda en el formulario. 4. Seleccionar la opción "Adicionar" que se muestra en la parte inferior del formulario.
	EC 1.2: El usuario deja campos vacíos.	El sistema muestra un mensaje especificando que debe rellenar el campo que dejo en blanco.	1. Seleccionar la pestaña "Gestionar solicitudes" en el menú de la izquierda. 2. Seleccionar dentro de esta pestaña la opción "Adicionar solicitud". 3. Insertar los datos en los campos, según se corresponda en el formulario. 4. Seleccionar la opción "Adicionar" que se muestra en la parte inferior del formulario.
	EC 1.3: El usuario introduce alguno de los datos requeridos de manera incorrecta.	El sistema muestra un mensaje indicando que existen datos incorrectos.	1. Seleccionar la pestaña "Gestionar solicitudes" en el menú de la izquierda. 2. Seleccionar dentro de esta pestaña la opción "Adicionar solicitud". 3. Insertar los datos en los campos, según se corresponda en el formulario. 4. Seleccionar la opción "Adicionar" que se muestra en la parte inferior del formulario.
	EC 1.4: El usuario carga un documento en un formato distinto al PDF.	El sistema muestra un mensaje indicando que los documentos a asociar con la solicitud deben estar en formato PDF.	1. Seleccionar la pestaña "Gestionar solicitudes" en el menú de la izquierda. 2. Seleccionar dentro de esta pestaña la opción "Adicionar solicitud". 3. Insertar los datos en los campos, según se corresponda en el formulario. 4. Seleccionar la opción "Adicionar" que se muestra en la parte inferior del formulario.
SC 2: "Actualizar Solicitud".	**EC 2.1:** El usuario modifica los datos de manera correcta.	El sistema actualiza los datos correspondientes a la solicitud modificada y envía una notificación al usuario que confirma que la acción se ejecutó correctamente.	1. Seleccionar la pestaña "Gestionar solicitudes" en el menú de la izquierda. 2. Seleccionar dentro de esta pestaña la opción "Mis solicitudes". 3. Seleccionar el ícono "Actualizar" de la solicitud, en la columna Opciones, de la tabla en que se muestra la lista de solicitudes. 4. Insertar datos en los campos a actualizar. 5. Seleccionar la opción "Actualizar" que se muestra en la parte inferior del formulario.

	EC 2.2: El usuario deja campos vacíos.	El sistema muestra un mensaje especificando que debe rellenar el campo que dejo en blanco.	1. Seleccionar la pestaña “Gestionar solicitudes” en el menú de la izquierda. 2. Seleccionar dentro de esta pestaña la opción “Mis solicitudes”. 3. Seleccionar el ícono “Actualizar” de la solicitud, en la columna Opciones, de la tabla en que se muestra la lista de solicitudes. 4. Insertar datos en los campos a actualizar. 5. Seleccionar la opción “Actualizar” que se muestra en la parte inferior del formulario.
	EC 2.3: El usuario introduce datos a actualizar incorrectos.	El sistema muestra un mensaje indicando que existen datos incorrectos.	1. Seleccionar la pestaña “Gestionar solicitudes” en el menú de la izquierda. 2. Seleccionar dentro de esta pestaña la opción “Mis solicitudes”. 3. Seleccionar el ícono “Actualizar” de la solicitud, en la columna Opciones, de la tabla en que se muestra la lista de solicitudes. 4. Insertar datos en los campos a actualizar. 5. Seleccionar la opción “Actualizar” que se muestra en la parte inferior del formulario.
	EC 2.4: El usuario actualiza los documentos asociados a la solicitud cargando uno o varios documentos en formato distinto al PDF.	El sistema muestra un mensaje indicando que los documentos a asociar con la solicitud deben estar en formato PDF.	1. Seleccionar la pestaña “Gestionar solicitudes” en el menú de la izquierda. 2. Seleccionar dentro de esta pestaña la opción “Mis solicitudes”. 3. Seleccionar el ícono “Actualizar” de la solicitud, en la columna Opciones, de la tabla en que se muestra la lista de solicitudes. 4. Insertar datos en los campos a actualizar. 5. Seleccionar la opción “Actualizar” que se muestra en la parte inferior del formulario.
SC 3: “Mostrar Solicitud”.	**EC 3.1:** Mostrar detalles de una solicitud.	El sistema muestra los datos y documentos asociados a la solicitud seleccionada.	1. Seleccionar la pestaña “Gestionar solicitudes” en el menú de la izquierda. 2. Seleccionar dentro de esta pestaña la opción “Mis solicitudes”. 3. Seleccionar el ícono “Detalles” de la solicitud, en la columna Opciones, de la tabla en que se muestra la lista de solicitudes.
SC 4: “Eliminar Solicitud”.	**EC 4.1:** Eliminar una solicitud.	El sistema elimina la solicitud seleccionada con todos sus datos y documentos asociados.	1. Seleccionar la pestaña “Gestionar solicitudes” en el menú de la izquierda. 2. Seleccionar dentro de esta pestaña la opción “Mis solicitudes”. 3. Seleccionar el ícono “Detalles” de la solicitud, en la columna Opciones, de la tabla en que se muestra la lista de solicitudes.

- **Descripción de las Variables**

Tabla. 9 Descripción de las variables.

No.	Nombre del Campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Puede ser cualquier combinación de 255 caracteres.
2	Cantidad de copias	Campo numérico	No	Debe ser un valor numérico entero mayor que cero.
3	Consumo de recursos	Lista desplegable	No	Debe seleccionar una de las opciones definidas en el sistema.
4	Documentos	Campo file (Examinar)	No	Debe contener documentos en formato “PDF”.

- **Matrices de datos**

Adicionar solicitud

Tabla. 10 Matriz de datos. Adicionar solicitud

Id. Del escenario	Escenario	Nombre	Cantidad de copias	Consumo de recursos	Documentos	Respuesta del sistema	Resultado de la prueba
EC 1.1	El usuario inserta los datos requeridos de manera correcta.	V/ (Planilla de actas)	V/ (30)	V/ (Del Área)	V/ (“Planilla.pdf”)	El sistema adiciona una nueva solicitud y envía una notificación al usuario que confirma que su solicitud se creó correctamente.	Satisfactorio
EC 1.2	El usuario deja campos vacíos.	I/ (“vacío”)	V/ (1)	V/ (Personal)	V/ (“Planilla.pdf”)	El sistema muestra un mensaje especificando que debe rellenar el campo que dejo en blanco.	Satisfactorio
EC 1.3	El usuario introduce alguno de los datos requeridos de manera incorrecta.	V/ (Planillas de actas)	I/ (-2)	V/ (Del Área)	V/ (“Planilla.pdf”)	El sistema muestra un mensaje indicando que existen datos incorrectos.	Satisfactorio

EC 1.4	El usuario carga un documento en un formato distinto al PDF.	V/ (Planillas de actas)	V/ (30)	V/ (Del Área)	I/ ("Planilla.docx")	El sistema muestra un mensaje indicando que los documentos a asociar con la solicitud deben estar en formato PDF.	Satisfactorio

Actualizar solicitud

Tabla. 11 Matriz de datos. Actualizar solicitud

Id. Del escenario	Escenario	Nombre	Cantidad de copias	Consumo de recursos	Documentos	Respuesta del sistema	Resultado de la prueba
EC 2.1	El usuario modifica los datos de manera correcta.	V/ (Planilla de actas)	V/ (30)	V/ (Del Área)	V/ ("Planilla.pdf")	El sistema modifica los datos de la solicitud y muestra un mensaje confirmando el éxito de la operación.	Satisfactorio
EC 2.2	El usuario deja campos vacíos.	I/ ("vacío")	V/ (1)	V/ (Personal)	V/ ("Planilla.pdf")	El sistema muestra un mensaje indicando que se deben especificar todos los campos para crear una solicitud.	Satisfactorio
EC 2.3	El usuario introduce datos a actualizar incorrectos.	V/ (Planillas de actas)	I/ (-2)	V/ (Del Área)	V/ ("Planilla.pdf")	El sistema muestra un mensaje indicando que existen datos incorrectos.	Satisfactorio
EC 2.4	El usuario actualiza los documentos asociados a la solicitud cargando uno o varios documentos en formato distinto al PDF.	V/ (Planillas de actas)	V/ (30)	V/ (Del Área)	I/ ("Planilla.docx")	El sistema muestra un mensaje indicando que los documentos a asociar con la solicitud deben estar en formato PDF.	Satisfactorio

Mostrar solicitud

Tabla. 12 Matriz de datos. Mostrar solicitud

Id. Del escenario	Escenario	Nombre	Cantidad de copias	Consumo de recursos	Documentos	Respuesta del sistema	Resultado de la prueba
EC 3.1	Mostrar detalles de una solicitud.	NA	NA	NA	NA	El sistema muestra los datos y documentos asociados a la solicitud seleccionada.	Satisfactorio

Eliminar solicitud

Tabla. 13 Matriz de datos. Eliminar solicitud

Id. Del escenario	Escenario	Nombre	Cantidad de copias	Consumo de recursos	Documentos	Respuesta del sistema	Resultado de la prueba
EC 4.1	Eliminar una solicitud.	NA	NA	NA	NA	El sistema elimina la solicitud seleccionada con todos sus datos y documentos asociados.	Satisfactorio

3.4.4 Resultados de la aplicación de las pruebas.

Como parte de la ejecución de las pruebas de caja negra se realizaron 3 iteraciones de pruebas que se representan en la figura 11. En la primera se identificaron 15 no conformidades, clasificadas en 13 no significativas y 2 significativas. Una vez corregidas, se procedió a realizar una segunda iteración, en la que se identificaron 4 nuevas no conformidades de tipo no significativas. Finalmente se realizó una última iteración en la que no se encontraron deficiencias, razón por la que se definió no realizar más iteraciones.

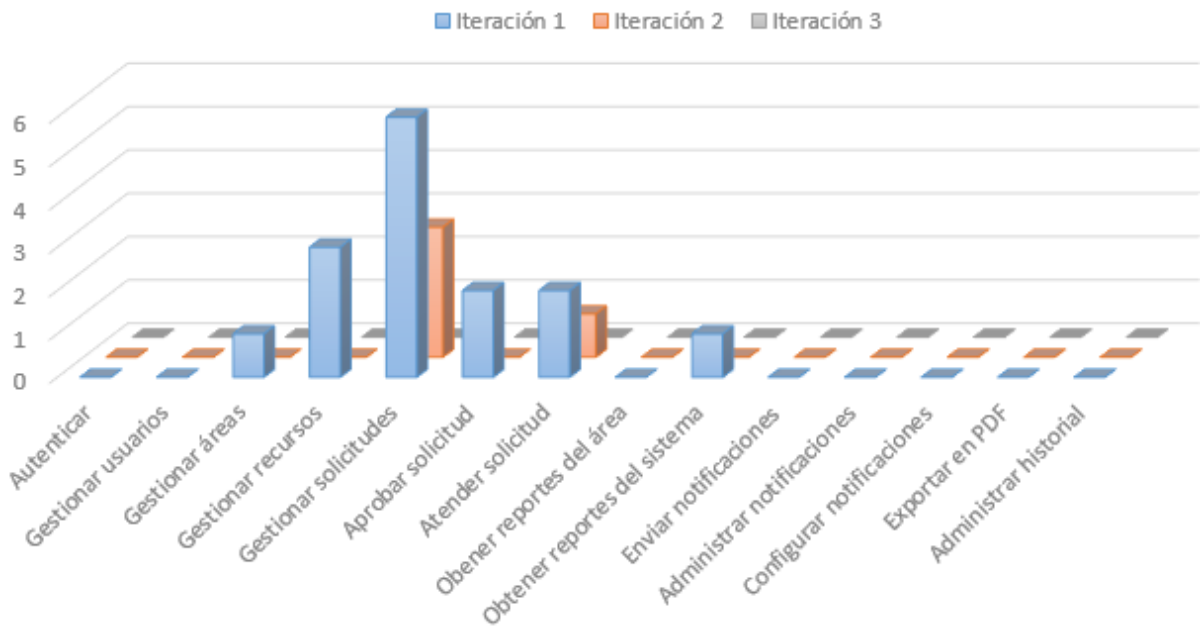

Fig. 11 Resultados de las iteraciones de las pruebas funcionales.

Con el desarrollo de las pruebas de aceptación se detectaron un total de 4 no conformidades de clasificación no significativa, relacionadas con los CU Gestionar solicitudes y Atender solicitud, las cuales fueran resueltas una vez que los especialistas del VDA informaron de estas a los desarrolladores del sistema. Como resultado de estas pruebas se emitió un acta de aceptación de la propuesta de solución por parte del usuario final y los desarrolladores (Ver anexo 3).

Para la realización de las pruebas de carga se utilizó la herramienta JMeter en su versión 2.3.1, mediante la cual se validó el RnF3 que plantea que el tiempo de respuesta promedio del sistema no debe superar los 5 segundos con 200 usuarios conectados simultáneamente. El resultado de estas pruebas se presenta en la figura 12, donde se muestra de las últimas 30 peticiones por ejemplo, el tiempo de respuesta en milisegundos para cada una en la columna "*Tiempo de muestra (ms)*", los errores detectados por la herramienta durante el proceso de petición y respuesta, en la columna "*Status*", el tiempo medio de respuesta para la muestra de 200 peticiones simultáneas (Media) y el valor de la desviación entre el mayor y el menor tiempo de respuesta (Desviación).

Muestra #	Start Time	Thread Name	Label	Tiempo de Muestra (ms)	Status	Bytes
170	23:58:15.588	Grupo de Hilos 1-166	/mis_solicitudes	4463		6005
171	23:58:15.574	Grupo de Hilos 1-158	/mis_solicitudes	4315		6005
172	23:58:15.597	Grupo de Hilos 1-167	/mis_solicitudes	4378		6005
173	23:58:15.592	Grupo de Hilos 1-170	/mis_solicitudes	4489		6005
174	23:58:15.617	Grupo de Hilos 1-169	/mis_solicitudes	4369		6005
175	23:58:15.622	Grupo de Hilos 1-176	/mis_solicitudes	4344		6005
176	23:58:15.612	Grupo de Hilos 1-174	/mis_solicitudes	4558		6005
177	23:58:15.708	Grupo de Hilos 1-188	/mis_solicitudes	4634		6005
178	23:58:15.628	Grupo de Hilos 1-177	/mis_solicitudes	4752		6005
179	23:58:15.739	Grupo de Hilos 1-193	/mis_solicitudes	4667		6005
180	23:58:15.652	Grupo de Hilos 1-180	/mis_solicitudes	4970		6005
181	23:58:15.745	Grupo de Hilos 1-194	/mis_solicitudes	4939		6005
182	23:58:15.674	Grupo de Hilos 1-184	/mis_solicitudes	5212		6005
183	23:58:15.669	Grupo de Hilos 1-183	/mis_solicitudes	5273		6005
184	23:58:15.657	Grupo de Hilos 1-181	/mis_solicitudes	5378		6005
185	23:58:15.641	Grupo de Hilos 1-178	/mis_solicitudes	5402		6005
186	23:58:15.687	Grupo de Hilos 1-186	/mis_solicitudes	5374		6005
187	23:58:15.647	Grupo de Hilos 1-179	/mis_solicitudes	5496		6005
188	23:58:15.759	Grupo de Hilos 1-197	/mis_solicitudes	5430		6005
189	23:58:15.718	Grupo de Hilos 1-191	/mis_solicitudes	5504		6005
190	23:58:15.610	Grupo de Hilos 1-176	/mis_solicitudes	5514		6005
191	23:58:15.684	Grupo de Hilos 1-185	/mis_solicitudes	5693		6005
192	23:58:15.728	Grupo de Hilos 1-192	/mis_solicitudes	5954		6005
193	23:58:15.755	Grupo de Hilos 1-196	/mis_solicitudes	5937		6005
194	23:58:15.615	Grupo de Hilos 1-173	/mis_solicitudes	5920		6005
195	23:58:15.693	Grupo de Hilos 1-187	/mis_solicitudes	6062		6005
196	23:58:15.619	Grupo de Hilos 1-175	/mis_solicitudes	5955		6005
197	23:58:15.618	Grupo de Hilos 1-172	/mis_solicitudes	5952		6005
198	23:58:15.704	Grupo de Hilos 1-189	/mis_solicitudes	6240		6005
199	23:58:15.663	Grupo de Hilos 1-182	/mis_solicitudes	6295		6005
200	23:58:15.751	Grupo de Hilos 1-195	/mis_solicitudes	6223		6005

No. de Muestras 200 **Última Muestra** 6223 **Media** 4821 **Desviación** 1524

Fig. 12 Resultados de las pruebas de carga.

Conclusiones parciales

Durante el desarrollo de este capítulo se describieron los elementos referentes al diseño en términos de componentes así como la distribución física del sistema. Además se realizó el proceso de pruebas, para el cual se desarrollaron pruebas funcionales, pruebas de desempeño y pruebas de aceptación.

Para las pruebas funcionales se utilizó el método de Caja Negra, mediante la técnica Partición de equivalencia, que arrojó 15 no conformidades en una primera iteración, 4 en la segunda y cero en la tercera. En el caso de las pruebas de aceptación desarrolladas por el vicedecano de administración, los especialistas del VDA y los desarrolladores, arrojaron como resultado la detección de 4 no conformidades no significativas, que fueron satisfactoriamente resueltas.

Para las pruebas de desempeño se realizaron pruebas de carga al sistema mediante le herramienta JMeter en su versión 2.3.1, lo que constató el cumplimiento satisfactorio de las expectativas planteadas de mantener un tiempo de respuesta promedio inferior a los 5 segundos con 200 usuarios conectados simultáneamente, considerado este el peor de los casos a los que se enfrentará el sistema.

CONCLUSIONES

Con el estudio de los sistemas para la gestión de información, enmarcados en los sistemas para la gestión de impresiones existentes, se pudo constatar que estas soluciones no satisfacen la problemática, lo cual evidencia la necesidad de la presente investigación.

A partir de la realización del análisis del sistema para la gestión del proceso de impresiones, se seleccionaron las tecnologías, herramientas y metodología de desarrollo que se ajustan a la situación del negocio.

Como parte del proceso de diseño se obtuvieron los diagramas y artefactos definidos por la metodología de desarrollo OpenUP, lo cual contribuirá a que el sistema sea escalable en el futuro.

Para validar el correcto funcionamiento de la propuesta de solución, se realizaron pruebas funcionales en 3 iteraciones que arrojaron 15, 4 y 0 no conformidades respectivamente. Además se realizaron pruebas de carga y de aceptación con las cuales se pudo corroborar que el sistema satisface los requisitos funcionales y no funcionales planteados.

RECOMENDACIONES

- Desarrollar una versión superior del sistema que brinde la posibilidad de calcular el consumo de tinta de los documentos a imprimir.

REFERENCIAS BIBLIOGRÁFICAS

1. ABRAN, A. and J. W. MOORE. *Swebok. Guide to the Software Engineering Body of Knowledge*. Estados Unidos de América, IEEE Computer Society Professional Practices Committee, 2004. 0-7695-2330-7
2. ADOBE. *About Adobe PDF*, 2014. [Disponible en: http://www.adobe.com/products/acrobat/adobepdf.html
3. ALFONSO, R. L. and Y. C. GARCÍA. *Automatización del Servicio de Impresión de la Dirección de Información."*. Habana, Cuba, Universidad de las Ciencias Informáticas, 2009. 103. p.
4. ALMEIRA, A. *Arquitectura de Software: Estilos y Patrones*. Facultad de Ingeniería Argentina, Universidad Nacional De La Patagonia San Juan Bosco, 2007.
5. BALDUINO, R. *Introduction to OpenUP*, 2007.
6. BOOCH, G. *El lenguaje unificado de modelado uml 20 guia de usuario aprenda uml directamente de sus creadores* 2006.
7. COMUNITY, P. *Introduction to PgAdmin*, 2012. [Disponible en: https://www.pgadmin.org
8. COULIBALY, N. *An Opensource GIS tool for Integrated Water Resources Management (IWRM) in a basin*. Bombay, India, Indian Institute of Technology Bombay, 2011.
9. CUBA, C. G. D. L. R. D. *Resolución No. 60/11 Normas del Sistema de Control Interno*, 2011. [60/11].
10. DANGEL, A. D. *Gestión del Conocimiento: Una Herramienta Esencial para el Diseño de Sistemas de Información*, 2009. [Disponible en: http://www.econlink.com.ar/gestion-conocimiento
11. DAVIS, G. and J. OLSON. *Management Information Systems: Conceptual foundations, Structure and Development.* New York, McGrawhill, 1985.
12. DOUGLAS, K. and S. DOUGLAS. *PostgreSQL. A comprehensive guide to building, programming, and administering PostgresSQL databases.* United States of America, Sams Publishing, 2003. 791.
13. EGUILUZ, J. *Introducción a JavaScript* 2006.
14. FIGUEROA, R. G.; C. J. SOLÍS*, et al. Metodologías Tradicionales Vs. Metodologías Ágiles.* Universidad Técnica Particular de Loja, Ecuador, 2011.
15. GAONA, A. L. *El modelo Entidad- Relación* Mexico, 2012.
16. GARCIA, R. G. *"Analisis, diseño y desarrollo de un sitio web para la empresa creaciones sortijerito"*. Ecuador, ESPE Universidad de las Fuerzas Armanadas de Ecuador, 2014. 162. p.

17. GONZÁLEZ, D. A. H. *Diagramas de Casos de Uso del Negocio y del Sistema*. Habana, Cuba, Instituto Superior Politécnico Jose Antonio Hechavarría (CUJAE), 2013.
18. GONZÁLEZ, J. P.; F. J. DOMINGUEZ, et al. Pruebas de aceptación orientadas al usuario: contexto ágil para un proyecto de gestión documental. España, Universidad de Sevilla, 2014.
19. HERNÁNDEZ, R. A. and S. COELLO. *El proceso de investigación científica*. 2011. 110 p.
20. IEEE IEEE Recommended Practice for the Adoption of Computer-Aided Software Engineering (CASE) Tools, 1995.
21. JACOBSON, I.; G. BOOCCH*, et al. El proceso unificado de desarrollo de software*. Madrid, España, Pearson Educacion S.A., 2000. 464 p.
22. LARMAN, C. *UML y Patrones*. 2da. Prentice Hall, 2003.
23. LEWIS, W. *Software Testing and Continuous Quality Improvement.* Estados Unidos de América, Auerbach, 2005.
24. MICROSOFT. *Understanding the Excel .xls Binary File Format*, 2011a. [Disponible en: https://msdn.microsoft.com/en-us/library/office/gg615597%28v=office.14%29.aspx ---. *Understanding the Word .doc Binary File Format*, 2011b. [Disponible en: https://msdn.microsoft.com/en-us/library/office/gg615596%28v=office.14%29.aspx
25. MOREIRO, G. *Introducción al estudio de la información y la documentación.* Medellin, Colombia, Editorial de Antioquía, 1998.
26. MORENO, F. *DALI Sistema centralizado de gestión de la impresión.* España, 2014.
27. NAVARRETE, T. *El lenguaje JavaScrip*, 2007.
28. NETBEANS. ¿Qué es NetBeans? en., 2015.
29. PLEVA, J. *PHP: Hypertext Preprocessor*. Estados Unidos de América, University of Wisconsin-Platteville, 2013.
30. PONJUAN, G. *Gestión de la información en las organizaciones: principios, conceptos y aplicaciones.* Santiago de Chile, 1998.
31. POTENCIER, F.; R. WEAVER*, et al. Buenas prácticas oficiales de Symfony*. 2014.
32. PRESSMAN, R. S. *Ingeniería del Software*. 2008.
33. SALGADO, S. R.; C. H. RAZA*, et al. Aplicación de la metodologia OpenUP en el desarrollo del sistema de difusión de gestión del conocimiento de la ESPE*. Ecuador, SANGOLQUÍ, 2013.
34. SARASWAT, D. Cloud Printer: A Survey *International Journal of Information and Computation Technology*, 2014, 4.

35. SKRABA, M. I. *Los Sistemas de Gestión de Impresión*. Eslovenia, Universidad de Liubliana, Facultad de Informática y Ciencias de la Información, 2013.

36. SOMMERVILLE, I. *Ingeniería del Software*. Madrid, España, Pearson Education S. A., 2005.
37. TABARES, Z. E. M.; D. S. LÓPEZ*, et al. Aplicación web para la realización de estudios farmacocinéticos, versión 2.0*. *Revista Cubana de Informática Médica*. Habana, Cuba, 2013. 5.
38. WHITE, M. *Intelligence Management. Informations Mnagemente from Estrategies to Action*. London, 1985.
39. WITTMANN, O.-R. *OASIS Open Document Format for Office Applications (OpenDocument) TC*, 2012. [Disponible en: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office
40. WOODMAN, L. *Information Management in Large Organizations. Information Management from Estrategies to Action*. 1985.
41. ZANINOTTO, F. P. A. F. *Symfony 2. La guía definitiva*, 2011.

GLOSARIO DE TÉRMINOS

Configurabilidad: Describe la capacidad de un programa para modificarse según las necesidades del usuario mediante opciones que modifiquen la función para el uso o fin para el que se desarrolló.

Historial: Reporte que almacena las acciones que se realizan en el sistema de manera histórica.

Log: Registro oficial de eventos, durante un rango de tiempo en particular, que es usado para registrar datos o información sobre eventos en un dispositivo o en una aplicación.

Multiplataforma: Se utiliza el término para denominar a los programas, lenguajes de programación u otra clase de software que pueden brindar sus prestaciones funcionando sobre diversas combinaciones de hardware y software.

Notificación: Mensaje en forma de aviso que utiliza el sistema para confirmar a los usuarios la ejecución satisfactoria de sus acciones y para dar seguimiento al estado de sus solicitudes.

ORM: Objeto relacional de mapeo. Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional como motor de persistencia.

Recursos: Son los aseguramientos materiales que provee el vicedecano administrativo a cada área para realizar el proceso de gestión de impresiones.

Servidor: Software u ordenador que provee servicios a otros programas o equipos denominados clientes.

UML: Lenguaje Unificado de Modelado. Constituye un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

ANEXOS

Anexo 1: Entrevista a trabajadores del VDA de la facultad 6.

Somos estudiantes de la carrera de Ingeniería en Ciencias Informáticas de la UCI, para la realización de nuestro trabajo de diploma se ha realizado una investigación acerca del funcionamiento y características del proceso de impresiones en el VDA de la facultad 6. Consideramos que tiene información valiosa para el desarrollo de esta investigación, dado que usted pertenece a esta área y posee la experiencia necesaria para facilitarnos los detalles y particularidades de este proceso, por lo que solicitamos su colaboración y le informamos el carácter confidencial de sus respuestas.

Gracias de antemano.

Objetivo: Identificar las principales características del proceso de gestión de impresiones en el VDA, así como las principales deficiencias y necesidades que enfrenta este procedimiento actualmente.

1. ¿Existen reglas para el desarrollo del proceso de impresiones?
2. ¿Estas reglas satisfacen las necesidades de organización y de control del proceso?
3. ¿Qué deficiencias a su entender afectan el desarrollo satisfactorio del proceso?
4. ¿Existen otros procesos que se lleven a cabo en el VDA que se vean afectados por la manera en que se desarrolla actualmente el proceso de impresiones?
5. ¿Existe alguna manera de auditar los documentos o conocer las estadísticas del consumo de recurso asignados al proceso?
6. Referente a los recursos que se destinan para el proceso ¿Qué deficiencias usted considera que inciden negativamente en su control?
7. ¿De qué manera usted considera que se pudiera elevar el nivel del control de los recursos que se asignan para el proceso?

Anexo 2: Entrevista a especialistas de la dirección de Información de la UCI.

Somos estudiantes de la carrera de Ingeniería en Ciencias Informáticas de la UCI, para la realización de nuestro trabajo de diploma se ha realizado un estudio de las soluciones existentes que pudieran aportar en el desarrollo de la propuesta de solución para la problemática que nos ocupa. Consideramos que tiene información valiosa para el desarrollo de esta investigación, dado que usted interactuó directamente con el Sistema para la Gestión de Impresiones de la Dirección de Información desarrollado y desplegado en el año 2009, por lo que solicitamos su colaboración y le informamos el carácter confidencial de sus respuestas.

Gracias de antemano.

Objetivo: Resumir las principales características referidas al funcionamiento del Sistema para la Gestión de Impresiones de la Dirección de Información de la UCI, como parte de las soluciones existentes.

1. ¿El sistema actualmente está en explotación?
2. ¿Cómo funciona el negocio para el que fue concebido el sistema?
3. ¿Qué funcionalidades presentaba el sistema?
4. ¿Cuáles eran las principales ventajas que propiciaba el sistema?
5. ¿Los desarrolladores del sistema se encuentran en la UCI?
6. ¿El código fuente del sistema se conserva a pesar de que el sistema ya no se utilice?

Anexo 3: Acta de aceptación de la propuesta de solución.

UCI
Universidad de las Ciencias Informáticas

Vicedecanato de Administración Facultad 6

La Habana, 10 de Junio del 2015
"Año 57 de la Revolución".

ACTA DE ACEPTACIÓN

De una parte, el Vicedecanato de Administración de la facultad 6, en lo sucesivo VDA, de la Universidad de las Ciencias Informáticas, representado en este acto por: Msc. Omar Mar Cornelio, y de **otra parte** los estudiantes: Bernardo Hernández González y Osviel Rodríguez Valdés.

Primero: Que en cumplimiento de los requisitos funcionales han sido efectuadas las implementaciones correspondientes.

CONSIDERANDO: Que los hitos realizados han sido desarrollados con la calidad requerida y bajo las condiciones pactadas y aprobadas por **Las Partes**.

CONSIDERANDO: Que los hitos que se han ejecutado cumplen con los requerimientos establecidos

POR TANTO: Las Partes acuerdan formalizar mediante la presente Acta, la aceptación del producto:

Sistema para la gestión del proceso de impresiones del Vicedecanato de Administración de la facultad 6 de la Universidad de las Ciencias Informáticas.

Y para que así conste, se extiende la presenta **Acta** en dos (2) ejemplares, rubricados por **Las Partes**.

Bernardo Hdez Glez

Osviel Rodríguez Valdés

Entregan

Omar Mar Cornelio

Recibe

UCI Universidad de las Ciencias Informáticas
Decanato Facultad 6

1