

Universidad de las Ciencias Informáticas



Universidad de las Ciencias
Informáticas

TÍTULO: Solución informática para el control del registro de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

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

Autor (es):

Rosalina Ramírez Araluce.

Tutor (es):

Ing. Mairelys Boeras Velázquez.

Ing. Kilmeny Acuña Crespo.

Ing. José Alejandro García Calderón.

La Habana, Cuba

Junio de 2013.



“Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad.”

Albert Einstein.

Declaración de autoría

Declaro ser autora del presente trabajo de diploma y autorizamos al Centro de Gestión Universitaria (CENIA) de la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo.

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

Rosalina Ramírez Araluca.

Firma de la Autora.

Ing. Mairelys Boeras Velázquez.

Firma del Tutor.

Ing. Kilmeny Acuña Crespo.

Firma del Tutor.

Ing. José Alejandro García Calderón.

Firma del Tutor.

Datos de Contacto.**Tutor:** Ing. Mairelys Boeras Velázquez.

- Ingeniero en Ciencias Informáticas.
- Profesor instructor de seis años de experiencia.
- Actualmente cumple las funciones de asesora de mercadotecnia del Centro de Informatización Universitaria.

Correo electrónico: mboeras@uci.cu**Tutor:** Ing. Kilmeny Acuña Crespo.

- Ingeniero en Ciencias Informáticas.
- Especialista general con de cuatro años de experiencia.
- Actualmente se encuentra vinculado al departamento Sistema de Gestión Universitaria de la Universidad de las Ciencias Informáticas.

Correo electrónico: kacuna@uci.cu**Tutor:** Ing. José Alejandro García Calderón.

- Ingeniero en Ciencias Informáticas.
- Especialista general de tres años de experiencia.
- Actualmente se encuentra vinculado como líder de proyecto del Sistema Gestión Académica de Posgrado de la Universidad de Ciencias Informáticas.

Correo electrónico: jacalderon@uci.cu

Dedicatoria.

A Dios.

Por permitirme llegar a este momento tan importante de mi vida y darme salud para lograr mis objetivos. Por llenarme de fuerzas para levantarme cuando caigo y no dejarme vencer por las adversidades.

A mi madre.

Por ser mi motor impulsor y guía en todo momento. Por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero sobre todo: por su amor incondicional que me hace olvidar las asperezas de la vida. Por enseñarme siempre a dar lo mejor de mí, por ser mi mejor amiga. Por ser ella: indudablemente el mejor ser humano sobre la faz de la tierra.

A mis dos papás.

Por los ejemplos de perseverancia y constancia que me han dado, por el valor mostrado para salir adelante. Por siempre estar ahí para mí y por su amor.

A Luis Esteban y Osniel José, mis hermanos.

Por ser los niños más bellos del mundo, por llenarme de alegría la vida, y por hacerme sonreír siempre. Por llamarme Tata aunque pasen los años, por existir y darme el amor más puro del mundo.

A mi abuela.

Por ser única y especial, por ser un ejemplo a seguir. Por quitarme los castigos de la infancia, por sus caricias y por comprenderme siempre.

Agradecimientos.

A mí abuelito.

Gracias por su ejemplo intachable, por su sonrisa espléndida, por su cariño y dedicación.

A mis tíos Joel y Joan.

Gracias por ser siempre tan especiales conmigo y quererme como a una hija, por estar tan cerca aunque estemos tan lejos.

A mis tutores: Maírellys, Kílmeny y José Alejandro.

Gracias por su apoyo en el desarrollo del trabajo. Por su paciencia cuando las cosas parecen imposibles y por la confianza depositada en mí.

A mis compañeros Ariel, Israel, Yander, Orlando y Leandro.

Gracias por inventarle otra hora al día para dedicármela y ayudarme, por ser como hermanos, por cuidarme como si fuera una niña. Por soportarme sin protestar.

A Geidy, Alisnay y Saeret.

Gracias por su amistad durante estos 5 años, por apoyarme en los malos momentos y hacer grandiosos los buenos. Por enseñarme a conocer un mundo mejor que era invisible a mis ojos.

A mis familiares.

Gracias a todos los miembros de mi familia que me brindan su amor y su ayuda, sin ustedes no sería posible este logro.

A todos los que de una forma u otra han aportado un granito de arena en la realización de este trabajo. Gracias

Resumen

El Grupo de Servicios Legales (GSL) de la Dirección General de Producción (DGP) de la Universidad de las Ciencias Informáticas (UCI), no cuenta con una solución de software que permita el control del registro de la propiedad intelectual (PI) generada en la Universidad. La carencia de una solución informática, trae como resultado que la gestión del registro de la propiedad intelectual sea procesada mediante herramientas ofimáticas, como el Excel, lo que hace engorrosa la tarea de obtener los consolidados que con frecuencia solicita la DGP. Por tanto, se ve afectada la inmediatez del proceso de registro de la propiedad intelectual en la UCI, además del manejo y visibilidad de la información. Por ello, la presente investigación tiene como objetivo desarrollar un sistema que facilite el proceso de registro de la propiedad intelectual generada en la Universidad de las Ciencias Informáticas, siguiendo las normas establecidas para la Universidad y utilizando la arquitectura del Sistema de Gestión Universitaria. Para darle cumplimiento al objetivo anteriormente descrito, se realiza un estudio a nivel nacional e internacional, con el fin de conocer cómo se realiza el proceso de registro de los bienes intangibles, lo que ayuda conceptualmente al desarrollo de la solución de software que se desea implementar. Se identifican los requisitos funcionales y no funcionales que debe cumplir la solución informática que se desarrolla en la presente investigación. Se efectúa el diseño e implementación de la misma, así como la ejecución de las pruebas correspondientes que corroboran la corrección del módulo que se implementa. Por tanto, con el desarrollo del Sistema de Registro de la propiedad intelectual en la UCI se mejora dicho proceso en la Universidad.

Palabras clave: bien intangible, propiedad intelectual, sistema de registro.

Índice de contenidos

Introducción	1
Capítulo 1: Fundamentación Teórica.....	6
1.1 Introducción.....	6
1.2 Conceptos asociados al dominio del problema.....	6
1.3 Formas de proteger la propiedad intelectual	6
1.4 Análisis de sistemas similares.....	10
1.4.1 Ámbito internacional.....	10
1.4.2 Ámbito nacional.....	14
1.5 Marco legal de la investigación.....	15
1.6 Lenguajes, tecnologías, metodología y herramientas a utilizar en el desarrollo del Sistema..	16
1.6.1 Lenguajes de programación.....	16
1.6.2 Tecnologías.....	17
1.6.3 Sistema gestor de bases de datos: PostgreSQL 8.4.....	20
1.6.4 Servidor web: Apache 2.2	20
1.6.5 Herramientas	21
1.6.6 Modelado.....	22
1.6.7 Proceso de desarrollo de software	23
1.7 Conclusiones.....	26
Capítulo 2: Propuesta de solución.....	28
2.1 Introducción	28
2.2 Modelado del negocio.....	28
2.3 Modelado de procesos del negocio	29
2.4 Reglas de negocio.....	29
2.5 Propuesta del Sistema	30
2.6 Ingeniería de requisitos	31

2.7 Arquitectura del sistema	37
2.8 Modelo de datos	41
2.9 Diagrama de despliegue	42
2.10 Conclusiones	42
Capítulo 3: Implementación y pruebas del sistema.....	43
3.1 Introducción	43
3.2 Integración con otros módulos del Sistema de Gestión Universitaria	43
3.3 Integración entre módulos.....	44
3.4 Estándares de codificación.....	44
3.4.1 Identación, llaves de apertura y cierre y tamaño de las líneas.....	44
3.4.2 Conversión de nomenclatura.....	45
3.4.3 Estructuras de control.....	46
3.4.4 Documentación.....	46
3.4.5 Buenas prácticas	46
3.5 Estándares del diseño.....	47
3.6 Estrategias de validación de requisitos.....	47
3.7 Pruebas	47
3.7.1 Pruebas de integración	48
3.7.2 Pruebas de rendimiento y resistencia (stress).....	48
3.7.3 Pruebas de aceptación.....	50
3.7.4 Pruebas funcionales.....	50
3.8 Conclusiones.....	53
Conclusiones generales	54
Recomendaciones	55
Glosario de términos:	56
Bibliografía referenciada	59
Bibliografía consultada.....	63

Anexos	64
---------------------	----

Índice de tablas

Tabla 1. Requisitos funcionales del sistema.	34
Tabla 2. Requisitos no funcionales del sistema.	35
Tabla 3. Especificación del requisito funcional 'modificar monto por tipo de obra'.	37
Tabla 4. Prueba de integración con el módulo Seguridad.	48
Tabla 5. Caso de prueba para el RF_22 `mostrar entidad externa`.	52

Índice de figuras

Figura 1. Modelado del proceso.	29
Figura 2 Arquitectura Cliente - Servidor.	38
Figura 3. Patrón Modelo - Vista - Controlador.	39
Figura 4. Modelo físico de datos.	41
Figura 5. Diagrama de despliegue.	42
Figura 6. Ejemplo de indentación, llaves de apertura y cierre y tamaño de las líneas.	44
Figura 7. Ejemplo de conversión de nomenclatura.	45
Figura 8. Ejemplo de Conversión de nomenclatura para clases.	45
Figura 9 . Ejemplo de conversión de nomenclatura para funciones.	45
Figura 10. Ejemplo de estructuras de control.	46
Figura 11. Ejemplo de documentación.	46
Figura 12. Ejemplo de buenas prácticas.	47
Figura 13. Prueba de resistencia para una muestra de 500 usuarios.	49
Figura 14. Prueba de resistencia para una muestra de 20 usuarios.	50
Figura 15. No conformidades del sistema.	53

Introducción

Con el nacimiento de la imprenta, en el año 1450, se inicia la producción y venta de obras literarias, lo que produjo un sistema de privilegios para los impresores y un derecho de censura para los más poderosos de la época, dejando a los creadores de las obras sin reconocimiento. Con el decursar de los años, estos privilegios se modificaron dando paso al inicio de la propiedad intelectual (PI) (1). La Organización Mundial de la Propiedad Intelectual (OMPI) es el organismo dentro de las Naciones Unidas encargado de dar a conocer adecuadamente la propiedad intelectual y de cultivar su respeto a escala mundial. La propiedad intelectual es definida por la OMPI como: "cualquier propiedad que, de común acuerdo, se considere de naturaleza intelectual y merecedora de protección, incluidas las invenciones científicas y tecnológicas, las producciones literarias o artísticas, las marcas y signos distintivos, los dibujos y modelos industriales y las indicaciones geográficas" (2).

La propiedad intelectual representa uno de los bienes intangibles más importantes en el mundo pues tiene como objetivo dar protección jurídica a estos bienes, y con ello, al esfuerzo realizado en su proceso de creación. La necesidad de proteger los bienes intangibles ha crecido en los últimos tiempos debido a los progresivos avances tecnológicos, la competencia desleal y la globalización del mercado mundial. Mediante la propiedad intelectual se otorga derechos exclusivos sobre los bienes intangibles de los que es titular; y permitiendo comprar, vender, ceder bajo licencia, intercambiar y entregar gratuitamente el bien si se desea (2).

La propiedad intelectual se clasifica en dos categorías: el derecho de autor y la propiedad industrial. El derecho de autor abarca la protección de los derechos concedidos a los creadores por sus obras literarias o artísticas; y la propiedad industrial protege actividad del intelecto humano aplicada a la búsqueda de soluciones concretas de problemas específicos en el campo de la industria y del comercio (3).

Cada estado diseña su propio sistema de propiedad intelectual y define la validez territorial para sus obras. En Cuba, el Sistema Nacional de propiedad intelectual tiene como objetivo además, proteger convenientemente el patrimonio cubano de la competencia fraudulenta, logrando al mismo tiempo, brindar las correspondientes garantías legales a los inversores nacionales y extranjeros (4). Las instituciones responsables de realizar el registro de la propiedad intelectual en el territorio nacional son el Centro Nacional de Derecho de Autor (CENDA) y la Oficina Cubana de la Propiedad Industrial (OCPI) (5).

Gran parte de las creaciones que surgen en el mundo se obtienen del trabajo intelectual desarrollado en las universidades. La Universidad de las Ciencias Informáticas (UCI) dentro de su misión tiene: producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación (6). Para el cumplimiento de esta misión se realiza un gran trabajo intelectual en la Universidad, del cual

surgen obras que deben ser protegidas para evitar su uso inadecuado, algunas de ellas son: tesis de maestrías y doctorados, soluciones de software y marcas comerciales y productos. El registro de estos bienes intangibles ante las instituciones responsables del país es obligación del Grupo de Servicios Legales (GSL) de la Dirección General de Producción (DGP) de la UCI.

El proceso de registro de la propiedad intelectual no es un proceso complejo, pero requiere de la manipulación constante y organización de la información relacionada con las obras. Actualmente la información del proceso de registro, se asienta en herramientas ofimáticas como el Excel, y se hace engorrosa la tarea de obtener los consolidados que con frecuencia solicita la DGP. Aunque la información para los registros se lleva de forma centralizada en la oficina del GSL su consulta se dificulta pues el proceso se realiza utilizando la navegación por carpetas en la computadora donde se encuentra almacenada. Para conocer el estado de un proceso de registro de propiedad intelectual en la UCI, los autores de las obras y otros involucrados deben dirigirse a la oficina del GSL, ya que no se cuenta con un mecanismo factible para su constante divulgación. La falta de este mecanismo atenta contra la visibilidad de la información que interesa a autores e involucrados en el proceso, y el trabajo en la oficina se ve frecuentemente interrumpido, pues las personas se presentan en la misma para conocer el estado de dichos registros. Llevar el control de los períodos de vigencia de los registros de la propiedad intelectual es una tarea constante, ya que se necesita consultar frecuentemente las actas de registro para chequear que los periodos no venzan antes de ser renovados.

Teniendo en cuenta la situación problemática anteriormente planteada, se traza el siguiente **problema de la investigación**:

¿Cómo facilitar el proceso de registro de la propiedad intelectual en la Universidad de las Ciencias Informáticas?

Siendo el **objeto de estudio**: proceso de registro de la propiedad intelectual; enmarcado en el **campo de acción**: proceso de registro de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

El **Objetivo General** que se persigue es desarrollar un sistema que facilite el proceso de registro de la propiedad intelectual generada en la Universidad de las Ciencias Informáticas siguiendo las normas establecidas para la Universidad y utilizando la arquitectura del Sistema de Gestión Universitaria.

Se plantean como **objetivos específicos**:

- Elaborar el marco teórico asociado a la investigación científica.

- Diseñar la solución de software para el registro de la propiedad intelectual en la UCI a partir del proceso de desarrollo de software seleccionado.
- Implementar la solución de software para el registro de la propiedad intelectual en la UCI a partir del diseño desarrollado.
- Validar la solución de software desarrollada a partir de la aplicación de pruebas.

Se definen como **tareas de la investigación:**

- Análisis del estado del arte relacionado con sistemas o tendencias existentes vinculadas con el registro de la propiedad intelectual.
- Fundamentación de las tecnologías a utilizar en el desarrollo de la solución de software.
- Análisis de procedimientos, normativas y estándares vigentes que rigen el proceso de registro de la propiedad intelectual en la UCI.
- Descripción del proceso de registro de la propiedad intelectual en la UCI.
- Identificación de los requisitos funcionales y no funcionales para el desarrollo de la solución de software teniendo como base los procesos descritos.
- Diseño de las funcionalidades a implementar como parte de la solución de software.
- Diseño de las interfaces de usuario y la base de datos para la solución de software.
- Diseño de la integración de la solución con el Sistema de Gestión Universitaria (SGU).
- Implementación de las funcionalidades diseñadas.
- Implementación de la integración de la solución con el Sistema de Gestión Universitaria (SGU).
- Descripción de la estrategia de prueba a seguir para la validación de la solución de software desarrollada.
- Aplicación de pruebas a la solución de software.

Métodos científicos de la investigación: los métodos científicos utilizados para la investigación son los métodos empíricos y los métodos teóricos.

Métodos Empíricos: crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad. Posibilitan el conocimiento del estado del arte del fenómeno, su evolución en una etapa determinada, su relación con otros fenómenos (7).

- **Entrevista:** este método ha sido utilizado para obtener información adecuada y acertada a través de preguntas a los expertos en asesoría legal en registro de la propiedad intelectual en la Universidad de las Ciencias Informáticas, con el objetivo de conocer las inconformidades que han detectado en el proceso de registro y conocer potenciales requerimientos de software.

- **Observación bibliográfica:** este método se ha utilizado en la mayor parte de la investigación pues permite el análisis de la información mediante trabajos realizados por expertos, con el fin de conocer las normas, regulaciones, decretos y leyes que conforman el marco legal de la investigación, así como conocer el estado del arte para el desarrollo efectivo de la investigación.

Métodos Teóricos: permiten estudiar las características del objeto de investigación que no son observables directamente. Facilitan la construcción de modelos e hipótesis de investigación (7).

- **Análisis histórico-lógico:** este método se utilizó para investigar la información que se tiene hasta ahora sobre el registro de la propiedad intelectual y considerar los aspectos fundamentales para la realización de dicha investigación.
- **Analítico-sintético:** este método permitió procesar toda la información y dividir el problema en porciones más pequeñas con el fin de hacerlo más fácil de entender, y darle solución a cada porción del mismo por separado.

Justificación de la investigación:

La solución que se propone en esta investigación posibilita:

- La centralización más efectiva y mayor visibilidad de la información relacionada con la propiedad intelectual de la UCI.
- El aumento en la disponibilidad de la información relacionada al registro de propiedad intelectual.
- La disminución de interrupciones por concepto de notificación del estado de los registros de las obras a interesados.
- La agilidad en las búsquedas de información para la confección de reportes relacionados con los registros de la propiedad intelectual.
- Un mejor control de los períodos de vigencia de los registros de propiedad intelectual.

Estructura del contenido:

A continuación se muestra la estructura del presente trabajo, incluyendo una síntesis de los capítulos.

Capítulo1. Fundamentación Teórica: en este capítulo se describen los elementos teóricos que sirven de base a la investigación, y los conceptos fundamentales relacionados con la misma. Se describen los resultados obtenidos del estudio del estado del arte relacionado con sistemas utilizados para el registro de la propiedad intelectual. También se describen las herramientas y tecnologías utilizadas para el desarrollo de la solución que se propone como parte de la investigación.

Capítulo 2. Propuesta de solución: en el capítulo se describe el proceso para el registro de la propiedad intelectual, y el marco legal utilizado como base para el levantamiento de requisitos funcionales y no funcionales de la propuesta de solución. Se realiza el modelado de procesos de negocio y la especificación de cada una de las funcionalidades identificadas para la solución. Se describen la arquitectura que presenta el sistema y los patrones de diseño utilizados que se alinean con los definidos para el SGU, además se presenta el modelo de datos y el diagrama de despliegue de la solución.

Capítulo 3. Implementación y prueba: En el capítulo se describen los módulos que forman la solución y se describe la integración con el SGU. Se detallan los estándares utilizados en la escritura del código y en el diseño de la base de datos; además se explica la estrategia de pruebas utilizada para la validación del sistema y se muestran los resultados obtenidos de las mismas.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En este capítulo se definen conceptos asociados al dominio del problema para un mejor entendimiento del mismo. Comprende el análisis de sistemas similares que desarrollan el registro de la propiedad intelectual, así como, el marco legal de la investigación. Se caracteriza el proceso de desarrollo de software, lenguajes de programación y tecnologías que se utilizan para el desarrollo de software.

1.2 Conceptos asociados al dominio del problema

- **Bien intangible:** aquellos bienes que no poseen materialidad, que son creados por el conocimiento e intelecto humano. Ejemplos son las marcas, diseños industriales, obras de artes, por citar algunas.
- **Derecho:** conjunto de normas jurídicas que forman un sistema hermético, al punto que las soluciones hay que buscarlas en las propias normas (8).
- **Derecho exclusivo:** se asocia a un derecho o privilegio que tiene un individuo o institución para realizar “algo” que está prohibido o vedado para el resto (9).
- **Titular:** que consta en algún documento como beneficiario o propietario de “algo” (10).
- **Registro de la propiedad intelectual:** proceso que se lleva a cabo para asentar el otorgamiento del derecho exclusivo sobre un bien intangible a un titular.
- **Obra:** bien intangible que puede pertenecer a cualquiera de las categorías de propiedad intelectual. En la Universidad de las Ciencias Informáticas las obras susceptibles a registro son: audiovisuales, marcas, software, base de datos, de arte, científicas, diseño de comunicación social, fotográfica, literaria y auditorías DVD.
- **Organismo cotitular:** Organismo que interviene en conjunto con la UCI en la creación de la obra.
- **Entidad externa:** Organismo ajeno a la UCI vinculado a la creación de la obra.
- **Autor externo:** Creador de la obra que pertenece a una entidad externa.

1.3 Formas de proteger la propiedad intelectual

Existen cuatro formas fundamentales de proteger la propiedad intelectual, estas son: el registro, las licencias, los contratos y la confidencialidad.

– El registro

Entre los derechos que le confiere a su titular el registro de un bien intangible como puede ser una marca, una patente o un nombre comercial, por poner algunos ejemplos, están los de comprar, vender y ceder bajo licencia. En Cuba, están regulados los procedimientos a seguir para proteger la Propiedad Industrial y el

Derecho de Autor en sus diversas modalidades ante la Oficina Cubana de la Propiedad Industrial y el CENDA respectivamente.

¿Qué ventajas brinda un bien intangible debidamente registrado?

- El registro es sinónimo de legalización, por lo cual la propiedad intelectual está reconocida jurídicamente.
- Garantiza los derechos exclusivos sobre el bien intangible a los titulares de los mismos, lo que permite usarlo y autorizar a un tercero a hacerlo.
- Obstaculiza la competencia desleal.
- Brinda la posibilidad de acudir a los Tribunales y emplazar al infractor (4).

¿Qué desventajas ocasionan el uso y divulgación de un bien intangible sin registrar?

- Representa costos y riesgos elevados.
 - Se pueden violar los derechos exclusivos de un tercero.
 - Se puede incurrir en una prohibición de registro de marcas.
 - Proporciona gastos innecesarios por indemnización a aquellos a los que se les haya afectado su derecho.
 - Puede ocasionar la inutilización de etiquetas y el fracaso de Campañas Publicitarias.
 - Inversión inútil de tiempo y recursos (4).
- **Las licencias**

Los bienes intangibles, en particular los signos distintivos de una organización empresarial tales como las marcas y los nombres comerciales, pueden representar un considerable valor como parte del patrimonio de sus titulares. Además pueden ser objeto de negociación a través de acuerdos jurídicamente concertados entre aquellos y el interesado en hacer uso del signo distintivo. En el caso de las marcas y otros signos distintivos esta negociación se materializa a través de la forma jurídica denominada licencia de marca. Un ejemplo clásico de licencia de marca es la Franquicia, contrato por medio del cual el propietario de un producto, proceso o servicio permite que un operador o comerciante local establezca un negocio utilizando la marca que distingue el producto, proceso o servicio, o el nombre comercial de la empresa titular del bien intangible (4).

También se utilizan los llamados “acuerdos de licencia” mediante los cuales el licenciante otorga al licenciatarario un derecho limitado a fabricar, utilizar o vender el objeto a que se refiera la licencia, lo cual se hace contra una compensación o regalía. Estos acuerdos de licencia se refieren específicamente a las patentes, marcas o *know-how* (4).

En Cuba el Decreto-Ley No. 203, en sus artículos del 83.1 al 88, trata el tema de las licencias de marcas estableciendo los parámetros que deben ser observados por los titulares para negociar la marca registrada. Establece, además, disposiciones relativas al ejercicio de los derechos exclusivos y ejercicio de acciones referidas a la defensa de la marca objeto de negociación entre otros (4).

– Los contratos

Otra modalidad jurídica mediante la cual se pueden explotar los bienes de propiedad intelectual y en particular la industrial, es la llamada “transferencia de tecnología”, la cual se refiere a aquellos contratos mediante los cuales se trasmite el uso o se autoriza la explotación de patentes, marcas, nombres comerciales, modelos industriales; se suministran conocimientos técnicos y *know-how*, se suministra ingeniería para la construcción industrial o fabricación de productos o se prestan servicios de asistencia técnica, administración de empresas, consultoría, asesoría, etcétera (4).

Estos contratos pueden combinarse con otras modalidades de contratos en dependencia del objeto del negocio dado que el intercambio o transferencia de conocimientos técnicos se puede manifestar de diferentes formas y está presente en muchas de las actividades económicas de las empresas cubanas (4).

Ejemplos en Cuba donde se materializa esta combinación de modalidades en un mismo negocio, son los casos de los contratos de administración que son muy utilizados en el sector del turismo y por medio de los cuales se transmiten conocimientos y habilidades para el desarrollo de esta actividad por parte de una entidad extranjera con suficiente experiencia y prestigio en el giro (4).

También se combinan diversas modalidades en los contratos de servicios de las organizaciones consultoras, que en estos tiempos están tomando fuerza a escala mundial y Cuba no es una excepción de esta tendencia generalizada.

– La confidencialidad

La confidencialidad está dirigida a proteger aquella información que se considera “secreta” o que no es de conocimiento general de la industria y que confiere a su dueño la ventaja sobre sus competidores. Esta

clase de información se denomina “secreto industrial” o “secreto comercial” y puede ser una fórmula, un patrón, un método, programa o técnica, procesos o compilaciones de información que le confieren una ventaja competitiva a la empresa de quien los posee (4).

Para la protección de los secretos comerciales o industriales es fundamental establecer un programa de protección y seguridad de la información a todos los niveles de una empresa que no solo se limita a secretos comerciales sino que también incluye cualquier otro tipo de información confidencial dentro de la gestión empresarial. Esto implica la adopción de una política oficial en materia de seguridad y protección de la información (4).

Entre los principios generales del Sistema Nacional de Propiedad Intelectual de Cuba (SNPI), está regulado el de implementar mecanismos y procedimientos que preserven el carácter de información oficial clasificada o de información oficial limitada de los conocimientos e informaciones técnicas, de producción, comerciales, económicas, de mercado, de los modelos, marcas y otros signos distintivos en proyecto de protección a fin de proteger su valor como activo comercial (4).

La necesidad de fortalecer las bases legales de la propiedad intelectual en todas sus modalidades, se hace cada vez más evidente a partir del papel preponderante que están adquiriendo a escala mundial los bienes intangibles dentro de la nueva era de las tecnologías de la información y las comunicaciones. En estos momentos, en Cuba, donde el objetivo primordial de todas las organizaciones empresariales es encaminarse hacia el perfeccionamiento empresarial, la propiedad intelectual pasa a desempeñar un rol de suma importancia para alcanzar este fin. Se toma en cuenta, que este proceso requiere de una continua innovación de la gestión interna y externa de la empresa, así como la posibilidad de alcanzar de manera sistemática el óptimo desempeño para producir bienes y servicios más competitivos. La existencia de un acertado sistema de protección de la PI, conlleva también al desarrollo del comercio y las inversiones dentro del territorio nacional (4).

La cultura informacional de los empresarios, es otro elemento esencial en torno a la protección de este importante tipo de propiedad, ya que el desconocimiento de los mecanismos, procedimientos y disposiciones legales puede conspirar contra las oportunidades de negocios, la competitividad de los productos y tecnologías y el posicionamiento del mercado, puede propiciar también la competencia desleal. De las modalidades de la protección de la propiedad intelectual, la presente investigación se enmarca solamente en el estudio de una de estas, el registro, por ser el proceso base e inicial de la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

1.4 Análisis de sistemas similares

En la actualidad existen a nivel mundial diversos sistemas de gestión para los servicios jurídicos, dentro de estos asuntos legales se encuentra el registro de la propiedad intelectual, que contrario a otros servicios jurídicos, no ha tenido un gran número de sistemas informáticos que sirvan de apoyo al proceso.

1.4.1 Ámbito internacional

- *SafeCreative*

SafeCreative es la primera plataforma de registro, información y gestión de la propiedad intelectual para la realidad digital creada en Zaragoza, España. Al registrar una obra, *SafeCreative* proporciona información, seguridad y autogestión de los derechos de propiedad intelectual (11).

- **Información:** información en tiempo real de la propiedad intelectual, tanto para personas, a través de interfaz web y etiquetas informativas, como para programas y sistemas, a través de la tecnología semántica desarrollada por *SafeCreative: Semantic Copyright* (11).
- **Seguridad**
 - Para los autores: la prueba de autoría de un registro con garantías tecnológicas irrefutables, depositando de la obra registro de múltiples huellas digitales y doble sellado de tiempo.
 - Para los usuarios de las obras: certeza sobre la licencia y usos permitidos, seguridad frente a posibles futuros cambios de licencia por parte del autor.
- **Autogestión:** la tecnología e Internet hacen posible la autoproducción y la autodistribución por citar algunos, *SafeCreative* proporciona la primera plataforma que permite la autogestión independiente y directa para las nuevas realidades creativas de creadores que necesitan modelos diferentes de gestión de derechos (11).

Descripción del servicio

SafeCreative consiste en un registro o depósito de obras de propiedad intelectual en formato digital, al que se puede asociar, la política de cesión de uso de los derechos de explotación de la misma por medio de licencias/contratos predefinidos como: *Creative Commons*, Licencia Pública General (LPG) u otros sistemas. En este sentido, permite (11):

- La inscripción y publicación de la autoría o titularidad de la obra creativa: constituyendo una prueba frente a terceros que aleguen derechos sobre la misma.
- La inscripción y publicación de los términos y condiciones de reproducción, comunicación pública, distribución y demás derechos de explotación sobre la obra, facilitando la autogestión de estos derechos.
- La emisión de certificados de inscripción de autoría o titularidad de la obra, a fin de que llegado el caso pueda aportarse una prueba documental que acredite la fecha y hora del registro o depósito en *SafeCreative*.
- La emisión de certificados de inscripción de reserva o cesión de derechos, de modo que los usuarios tengan la garantía y seguridad de que el uso de los contenidos estaba sometido a determinadas condiciones aplicables a un momento dado.

El contrato suscrito entre las partes consiste en un depósito de carácter gratuito, por medio del cual el usuario, autor o titular de los derechos de explotación exclusivos de una obra de propiedad intelectual, aloja o deposita una reproducción de la obra exclusiva para su registro y custodia por parte de *SafeCreative*. Todo lo anteriormente descrito, se realiza de acuerdo a los términos del contrato. En caso de que el usuario así lo establezca, *SafeCreative* permite el acceso al contenido de la obra depositada; dígame título, nombre del autor, política de derechos y demás información facilitada en el proceso de registro, ya sea directamente o por medio de los buscadores, publicaciones y demás herramientas o funcionalidades que establezca *SafeCreative* (11).

Principales características de *SafeCreative*:

- Registro válido para cualquier opción de derecho de autor, ya sea *Copyleft* o *Copyright*.
- Prueba de autoría basada en: depósito de la obra, hash del fichero depositado, sellado de tiempo redundante de la fecha y hora del registro.
- Interfaz API abierto para registro y consulta desde sistemas externos.
- Marco tecnológico empleado: *Semantic Copyright*.
- Registro automático desde canales RSS.
- Contratación de licenciamiento directo con el autor.
- Registro gratuito (11).

Diferencia con los registros tradicionales

La diferencia entre el registro digital desarrollado por *SafeCreative* y el tradicional radica en la razón por la cual la prueba se considera legalmente válida. En los registros tradicionales la razón es la autoridad de la persona o entidad que certifica los registros. Su validez se basa en la autoridad conferida por la legislación del país en que se encuentre el titular o que la administración concede, a la organización o persona responsable del registro. Estos registros se nombran: dependientes de ministerios o notarios. El rigor de la prueba en los registros digitales resulta incuestionable por la tecnología empleada para su creación. No necesita una autoridad delegada por ninguna administración. Se trata de una prueba independiente del propio registro, que puede hacerse valer con independencia de éste. La identificación e integridad del fichero registrado, así como de la fecha de su registro, son irrefutables por los dos elementos técnicos probatorios:

- **Identidad del fichero:** en el caso de *SafeCreative* queda garantizada por una triple comprobación contra tres hashes o huellas digitales calculadas sobre el fichero registrado: MD5, SHA1 y SHA512 (11).
- **Fecha de registro:** garantizada por un doble sellado de tiempo redundante, que realiza la autoridad de certificación de *SafeCreative* y una autoridad de certificación externa homologada por la administración pública de Zaragoza (11).

Por todo lo anteriormente expuesto, se conoce que *SafeCreative* se rige bajo las normas y especificaciones de Zaragoza, España. Facilita el registro de la propiedad intelectual de esta región, actuando como intermediario entre el titular de la obra y la entidad encargada de llevar a cabo el proceso de registro. No necesita una autoridad delegada por ninguna administración para otorgarle validez al registro de la obra. Como consecuencia, no se utiliza este sistema para solucionar la situación problemática que enfrenta la UCI. Además para lograr que el registro sea válido, todo el proceso debe seguir los requerimientos cubanos. Por cuanto, debe ser realizado ante el Centro Cubano de Derecho de Autor o la Oficina Cubana de Propiedad Industrial, este sistema no lo garantiza debido a que se rige por el sistema legal de España.

- ***Semantic Copyright***

Es un sistema tecnológico que proporciona información semántica de los derechos de propiedad intelectual de obras en formato digital, a las que identifica por su propia morfología. El proyecto nace en mayo de 2009 tras su presentación en la conferencia *Role of Copyright Registries*, en español reglas de registro reservado, en *Global Digital Networks*, su primera versión fue publicada y liberados sus derechos de uso e

implementación, en marzo de 2011. Su desarrollo ha sido realizado por el registro de propiedad intelectual *SafeCreative*, el departamento de Inteligencia Artificial de la Universidad Politécnica de Madrid y Dialnet, con el apoyo Ministerio de Industria, Turismo y Comercio de España. (12)

Proporciona un marco basado en información semántica, que permite a los sistemas telemáticos consultar de forma automatizada derechos de propiedad intelectual de los ficheros que transmiten los contenidos como textos, imágenes, música, vídeo, programas de ordenador, entre otros. Los tres subsistemas que componen el marco tecnológico que comprende *Semantic Copyright* son:

- **Identificación de la obra:** el sistema externo interroga a través de cualquier nodo adscrito a *Semantic Copyright* sobre los derechos de autor de una obra. La identificación se basa en el análisis de la propia morfología del fichero digital que contiene la obra, y no es necesario que el mismo incluya metadatos o sistemas de etiquetado. La versión 1.0 emplea para el reconocimiento morfológico de ficheros huellas digitales: MD5, MD4 y las de la familia *Secure Hash Algorithm*.
 - **Consulta:** el sistema externo puede dirigir la consulta a cualquier nodo que emplee el modelo *Semantic Copyright*.
 - **Respuesta:** la respuesta semántica sobre los derechos de propiedad intelectual está formada por dos componentes:
 - Respuesta breve con la información fundamental de los derechos de la obra: nombre del autor y características básicas de los permisos o limitaciones para uso y difusión de la obra. Esta información permite al sistema arbitrar decisiones automatizadas sobre la distribución o no de la misma.
 - Dirección web en la que se incluye información semántica detallada con una ontología específica para el dominio del tipo de obra, por ejemplo: literaria, técnica, música o vídeo.
- (12).

Por todo lo anteriormente planteado, se llega a la conclusión de que *Semantic Copyright* es un sistema tecnológico que permite identificar si una obra es una copia de la original por su morfología. Esta identificación se realiza mediante un marco tecnológico de información semántica que contiene *Semantic Copyright*, el cual permite consultar a los sistemas telemáticos los ficheros contenidos en el sistema. Luego comprueba la morfología del fichero seleccionado, en el subsistema identificación de la obra, perteneciente al marco tecnológico, a través de huellas digitales: MD5, MD4 y las de la familia *Secure Hash Algorithm*.

Debido a que se desea registrar la propiedad intelectual de la UCI y no identificar la veracidad digital de la obra, no se utiliza como solución de software para la presente investigación.

1.4.2 Ámbito nacional

Hasta el momento, no se encuentran referencias de algún sistema informático realizado en el territorio nacional que incluya dentro de sus funcionalidades el registro y control de la propiedad intelectual, este proceso se realiza de forma manual por las instituciones encargadas: Centro Nacional de Derecho de Autor y la Oficina Cubana de la propiedad intelectual.

1.4.2.1 En la Universidad de las Ciencias Informáticas

La Universidad de las Ciencias Informáticas, cuenta con un sitio web de la biblioteca, entre las facilidades que brinda este servicio se encuentra la publicación de las tesis de grado realizadas en diferentes cursos académicos. Permite de esta manera, que la información relacionada a estas tesis sea consultada con facilidad por los usuarios que interactúan con la biblioteca digital de la UCI. En las búsquedas realizadas se encontraron trabajos que se mencionan en lo sucesivo.

Tesis "Propuesta del proceso de gestión de la información de la propiedad intelectual en la Universidad de las Ciencias Informáticas"

Realizada en la Universidad de las Ciencias Informática en el curso 2009-2010 en la facultad 15, actual facultad 3. Propone el modelado de procesos y el desarrollo de un sistema informático que gestiona la información relacionada con la propiedad intelectual en la Universidad de las Ciencias Informáticas. La propuesta no cumple con los requerimientos del sistema de software que necesita la UCI en la actualidad, debido a que no se desea solamente gestionar la información de la propiedad intelectual, sino también la información del proceso de registro de la propiedad intelectual de la Universidad. Se utiliza como parte de la documentación que se estudia para llevar a cabo la presente investigación.

Tesis "Propuesta del proceso de la gestión de la información de los servicios legales en la empresa ALBET, S.A"

Realizada en la Universidad de las Ciencias Informáticas en el curso 2009-2010 en la facultad 15, actual facultad 3. Se propone la gestión de la información de los servicios legales en la empresa ALBET, S.A. para guiar los procesos y contribuir con la mejorara de la organización, el tratamiento y seguimiento de la información que en ella se genera. El trabajo va enfocado a la gestión documental y no a la información generada en el proceso de registro de las obras creadas en la UCI. Además en la actualidad el proceso de registro que la Universidad desarrollaba a través de la empresa ALBET S.A ya no se ejecuta, por lo que el

trabajo que se estudia no se ajusta a las necesidades actuales. Es por ello que no se puede utilizar como solución al problema que se plantea en la presente investigación.

Tesis "Implementación de algoritmos para la protección de la propiedad intelectual en imágenes digitales"

Realizada en la Universidad de las Ciencias Informáticas en el curso 2009-2010 en la facultad 10, actual facultad 1. Propone que se implementen de algoritmos con el objetivo de proteger a los autores junto a la fotografía. Adaptándose a las necesidades específicas del proyecto de Informatización de la Prensa con el fin de integrarse al producto Pyxel. Este trabajo, solamente abarca la protección de imágenes y no el proceso de registro de la propiedad intelectual de la Universidad. Debido a esto, no se utiliza como solución de software para la presente investigación. Sirve, como apoyo a la solución de software que se desea desarrollar, debido a que aporta información relevante en las bases teóricas de la investigación que se lleva a cabo en el presente trabajo.

Se concluye por tanto que, el estudio de los sistemas homólogos existentes y los trabajos desarrollados en la Universidad, no brindan una respuesta completa a las necesidades de la UCI en el proceso de registro de las obras que forman parte de su patrimonio. Además algunas cumplen las normas legales regentes en otro país y no en el territorio nacional.

1.5 Marco legal de la investigación

- IPP-1000:2008 Elaboración y aprobación de los procedimientos y lineamientos para la actividad productiva.
- IPP-4000:2008 Registro de la propiedad intelectual.
- Ley No. 14/1977, del Derecho de Autor, de la Asamblea Nacional.
- Decreto-Ley No. 68/1983 de las Invenciones, Descubrimientos Científicos, Modelos Industriales, Marcas y Denominaciones de Origen, del Consejo de Estado.
- Decreto-Ley No. 203/1999 de las Marcas y otros signos distintivos, del Consejo de Estado.
- Resolución No. 999/1983 de las Disposiciones Complementarias para la mejor aplicación y la Base de Datos de lo establecido en el Decreto-Ley No. 68/1983, de la Academia de Ciencias de Cuba.
- Resolución Conjunta No. 1/1999, sobre la Protección y Comercialización de los Programas de Computación y las Bases de Datos, del Ministerio de Cultura y el Ministerio de la Industria Sideromecánica.
- Resolución No. 33/2008, sobre el Sistema de Registro de producto de Software, del Ministerio de la Informática y las Comunicaciones.

- Resolución Rectoral No. 593/2007, Autorización al Director general de la Infraestructura Productiva a firmar la documentación que garantice los servicios y la protección de las áreas de la Infraestructura Productiva, del Rector de la Universidad de las Ciencias Informáticas.

1.6 Lenguajes, tecnologías, metodología y herramientas a utilizar en el desarrollo del Sistema

La solución de software, se desarrolla utilizando las tecnologías y herramientas usadas en el Sistema de Gestión Universitaria (SGU) al que se desea integrar. SGU, establece utilizar como lenguaje de programación PHP, GUUD como marco de trabajo para PHP, las bibliotecas JQuery de Javascript, PostgreSQL como gestor de base de datos, el proceso de desarrollo con enfoque ágil basado en el nivel 2 de CMMI y se hacen uso de las metodologías Scrum y XP.

1.6.1 Lenguajes de programación

Se considera lenguaje de programación, a cualquier lenguaje artificial que se utilice para definir una secuencia de instrucciones para su procesamiento, ejecutado por un ordenador o computadora. Existen distintas tipologías de lenguajes de programación, entre estas se encuentran: lenguaje de programación del lado del servidor y lenguaje de programación del lado del cliente.

1.6.1.1 Lenguaje de programación del lado del servidor PHP 5.3

PHP (*Hypertext Preprocessor*) es un lenguaje de código abierto, de alto nivel y especialmente se adapta al desarrollo de aplicaciones web. Dicho desarrollo, se incrusta en HTML (*HyperText Markup Language*) y se ejecutada en el servidor (13), se encuentra entre los lenguajes script más usados, debido a que presenta varias características importantes como:

- **Velocidad:** posibilita velocidad de ejecución y no crea demoras de procesamiento en el ordenador, ya que es un lenguaje interpretado que se ejecuta del lado del servidor, razones por las cuales no requiere demasiados recursos del sistema cliente.
- **Estabilidad:** utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- **Seguridad:** provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo .ini.
- **Simplicidad:** permite a los programadores generar código en el menor tiempo posible, los programadores con experiencia en C y C++ podrán utilizar PHP rápidamente.

PHP es completamente expandible y se compone en un sistema principal, un conjunto de módulos y una

variedad de extensiones de código. Generalmente se utiliza como módulo de Apache, lo que lo hace extremadamente veloz y al estar escrito en C permite que se ejecute rápidamente utilizando poca memoria. Puede interactuar con muchos motores de bases de datos tales como: MySQL, MS SQL, Oracle, Informix y PostgreSQL. Es código libre o abierto (*Open Source*), lo que significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan correctamente (14).

1.6.1.2 Lenguaje de programación del lado del cliente

1.6.1.2.1 JavaScript 1.6

Lenguaje interpretado, que no requiere compilación, funciona del lado del cliente y los navegadores se encargan de interpretar el código, no necesita que se instale ningún *framework*. Se incorpora en cualquier página web y se ejecuta sin la necesidad de instalar otro programa para su visualización. Es un lenguaje con muchas posibilidades, que se utiliza para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Permite crear diferentes efectos e interactuar con los usuarios. JavaScript es soportado por la mayoría de los navegadores como: Internet Explorer, Netscape, Opera, Mozilla Firefox (15).

1.6.1.2.2 HTML 4 (Hyper Text Markup Language)

Permite que se describa hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces que conducen a otros documentos o fuentes de información relacionadas y con inserciones multimedia. Se basa en especificar en el texto la estructura lógica del contenido, como títulos, párrafos de texto normal, enumeraciones, definiciones, citas; así como los diferentes efectos que se quieren dar como especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (16).

1.6.1.2.3 CSS 2 (Cascading Style Sheets)

En español, lenguaje de hojas de estilos, controla el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML (*eXtensible Hyper Text Markup Language*). Se conoce como la mejor forma de separar los contenidos y su presentación, es imprescindible para crear páginas web complejas.

Obliga a crear documentos HTML/XHTML bien definidos y con significado completo. Además mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes (17).

1.6.2 Tecnologías

1.6.2.1 Marco de trabajo: GUUD 1.0

GUUD se define como marco de trabajo que se utiliza en la presente investigación. Se desarrolla y adapta por el equipo de arquitectura del Centro de Informatización Universitaria (CENIA). El mismo integra a su vez los *frameworks* CodeIgniter 1.7.3 y JQuery en una sola infraestructura, razón por la cual posee las características de ambos en conjunto y además, esta integración incluye un conjunto de mejoras y modificaciones hechas específicamente para CodeIgniter.

– CodeIgniter 1.7.3

CodeIgniter, poderoso marco de trabajo, se utiliza en la creación de aplicaciones web usando PHP. Es de código libre. Facilita la utilización de librerías que ayudan al desarrollo rápido de aplicaciones web, además propone las pautas de desarrollo que se debe seguir para obtener provecho de una aplicación, reduciendo al mínimo la cantidad de código necesario para concebir una funcionalidad determinada. Se instala y usa con facilidad. Muchas de sus utilidades y modos de funcionamiento son opcionales, lo que permite una mayor libertad a la hora de desarrollar sitios web. Presenta diversas características importantes destacándose:

- **Versatilidad:** permite el trabajo en la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde solo se tiene un acceso por FTP para enviar los archivos al servidor y donde no se tiene acceso a su configuración.
- **Facilidad de instalación:** para subir CodeIgniter al servidor solamente se necesita una cuenta FTP y para su configuración solo se requiere la edición de un archivo y durante esta configuración no se necesita acceso a herramientas como la línea de comandos.
- **Flexibilidad:** es menos rígido que otros *frameworks*. Define una manera de trabajar específica; sin embargo, en muchos de los casos se puede seguir o no y sus reglas de codificación muchas veces se pueden saltar para que se trabaje más a gusto. Presenta módulos totalmente opcionales como el uso de plantillas.
- **Ligereza:** el núcleo de CodeIgniter es ligero, lo que permite que el servidor no se sobrecargue, interpreta o ejecuta grandes porciones de código. La mayoría de los módulos o clases que ofrece se cargan de manera opcional (18).

– JQuery

Librería de JavaScript, que facilita la forma de interactuar con los documentos HTML, el acceso a los objetos del DOM. Permite el manejo de eventos, el desarrollo de animaciones y agrega la interacción con la técnica AJAX páginas web. Ofrece una serie de funcionalidades que se basan en JavaScript, que posibilita grandes

resultados en la menor cantidad de tiempo y espacio posible. La característica principal de la biblioteca es que permite el cambio del contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX.

Consiste, en un conjunto de librerías integradas en un único archivo de varios kilobytes que permite, enriquecer estéticamente una página web. Proporciona funciones listas para usar; permite que facilite y estandarice la creación de aplicaciones, además posibilita que se obtenga información de los navegadores. Es compatible con los navegadores más usados a nivel internacional como Firefox desde su versión 3.6, Internet Explorer desde la versión 6.0, Safari desde la versión 5.0, Opera y Google Chrome (19).

Novedades o mejoras que incorpora el GUUD

A continuación se muestra una relación de las principales mejoras y modificaciones que incorpora el GUUD en su infraestructura.

Del lado del cliente:

- Se implementan una serie de widgets para utilizarlos de interfaz de algunos de los widgets base de jquery-ui, por ejemplo: el “*date*” y el “*tab*”. Ambos son interfaces de los widgets de mismo nombre de jquery-ui y el *popup* que es la interfaz del *dialog* de jquery-ui. Además de los ya mencionados, se implementan otros nuevos entre los que se encuentran: “*attach*”, menú, “*message*”, “*tooltip*”, “*form*”, que se construye con la unión de los plugins form de JQuery, el cual se utiliza para el envío de formularios AJAX y el “*validate*” se utiliza para validar formularios, “*grid*” el cual emplea como plugin el jqgrid, “*multiselect*” se usa para hacer selecciones múltiples, “*navbar*” crea las barras de navegación, “*tree*” utiliza la creación de árboles y el “*graph*” que usa la librería *Highchart* en la agrupación funcional Reportes (20).
- Contiene implementado un *plugin* a JQuery para el manejo de espacios de nombre e internacionalización (20).
- Se implementan funciones comunes para todo el sistema (contenidas en los archivos core.js y common.js) entre las que se destacan: “*loadIn*”, “*getDataJson*”, “*createSelect*”, “*isArray*”, “*isFunction*”, “*site_url*” (20).

Del lado del servidor, hechas a CodeIgniter (20):

- Se le implementa la programación orientada a aspectos.
- Se le agrega manejo de excepciones y mensajes.

- Se le implementa el IOC, *Inversion Of Control* o en español Inversión de Control, para la interacción entre módulos.
- Se le añade la característica de la modularidad o sea que una aplicación pueda dividirse en módulos. CodeIgniter no cuenta con esta posibilidad.
- Se incrementan los *plugins* export_pi que permiten exportar a los formatos: pdf, csv y xls; e import_pi, que brindan la posibilidad de importar desde archivos en formatos csv o xls.
- Se agregan, modifican y extienden los “*helpers*” o asistentes entre los que se encuentran (20):
 - **Agregados:** “*template*” que brinda la posibilidad de usar plantillas, característica que no posee CodeIgniter. Para esto se añadió también la librería *template*, “*assets*” utilizado para la integración en las vistas de JavaScript, CSS, imágenes y el *template*, “*grid*”, “*json*”
 - **Modificados:** “*form*”, “*array*”, “*security*”.

1.6.3 Sistema gestor de bases de datos: PostgreSQL 8.4

El sistema gestor de bases de datos (SGBD) relacionales orientadas a objetos conocido como PostgreSQL, se deriva del paquete Postgres. PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofrece control de concurrencia Multi-versión. Soporta casi toda la sintaxis SQL, cuenta con un amplio conjunto de enlaces con lenguajes de programación como PHP, C, C++, Java, Perl y Python (21). Es un sistema multiplataforma soportado por los principales sistemas operativos: Linux, Unix, Mac OS y *Windows*.

Presenta una documentación bien organizada, pública, libre y con comentarios de los usuarios. Soporta integridad referencial, la cual se utiliza para garantizar la validez de los datos de la base de datos (BD). Es altamente adaptable a las necesidades del cliente. Ofrece utilidades para limpieza de la base de datos (Vacuum) y almacenaje especial para tipos de datos grandes (TOAST). Utiliza una estrategia de almacenamiento de filas llamada MVCC para conseguir una respuesta mucho mejor en ambientes de grandes volúmenes (22).

1.6.4 Servidor web: Apache 2.2

Servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux), Windows y otras. Es un servidor altamente configurable se desarrolla por módulos. Además, permite a cualquier persona que posea experiencia en la programación de C o Perl que escriba módulos para que se realice una función determinada.

Apache es compatible con diferentes lenguajes script como PHP y Perl. Utiliza parte de Perl como soporte CGI y trabaja también con Java, posee el soporte que se necesita para tener páginas dinámicas. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Apache se configura para que ejecute un determinado script cuando ocurra un error en concreto. Posibilita un alto grado de configuración en la creación y gestión de registros, de este modo se puede tener un mayor control sobre lo que sucede en el servidor (23).

1.6.5 Herramientas

1.6.5.1 Herramienta de modelado: Visual Paradigm 8.0 Enterprise Edition

Herramienta CASE UML profesional, soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Contribuye a que se construya rápidamente aplicaciones de calidad y a un menor costo. Permite que se dibujen todos los tipos de diagramas de clases, código inverso; genera código desde diagramas y genera documentación. Esta herramienta de modelado, se utiliza para realizar los diagramas de procesos del negocio, utilizando la Notación para el Modelado de Proceso de Negocio (BPMN por sus siglas en inglés) (24).

1.6.5.2 Herramienta para el diseño de las interfaces de usuario: Evolus Pencil 1.3.5

Herramienta de prototipado libre y de código abierto con interfaz gráfica de usuario, que se instala con facilidad y crea maquetas de plataformas de escritorio populares. Evolus, brinda disponibilidad para todas las plataformas.

Características principales (25):

- Los proyectos se exportan en los formatos HTML, PNG, documento Openoffice.org, documento de Word y PDF.
- Instala plantillas definidas por el usuario.
- Permite operaciones de dibujo estándar: alinea, escala y rota.
- Permite la adición de los objetos externos.

1.6.5.3 Entorno de desarrollo integrado: NetBeans 7.1

Plataforma que permite el desarrollo de aplicaciones de software. Admite diferentes lenguajes de programación como C, C++, PHP y Java, mediante los cuales se pueden crear aplicaciones gráficas. Posibilita que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Las aplicaciones construidas a partir de módulos se pueden extender agregándole nuevos módulos, debido a que permiten ser desarrollados independientemente. Las aplicaciones

implementadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

NetBeans se considera un producto libre y gratuito. Sin restricciones de uso, que permite que los programadores puedan escribir, compilar, depurar y ejecutar programas. Además, presenta características muy valiosas para los desarrolladores como:

- Auto-completado y documentación de funciones PHP.
- Soporte para el control de versiones.
- Atajos de teclados (26).

1.6.5.4 Herramienta para administración de datos: PgAdmin III 1.10.0

Aplicación gráfica para el trabajo con el gestor de bases de datos PostgreSQL, de código abierto y se considera la más completa entre las de su tipo. Gestiona versiones a partir de PostgreSQL 7.3. Está escrita en C++, usa la librería gráfica multiplataforma *wxWidgets*, lo que permite que se pueda usar en Linux, *FreeBSD*, Solaris, Mac OS X y *Windows*.

PgAdmin III por su diseño, responde a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. Incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, por citar algunos. La conexión al servidor se hace mediante conexión TCP/IP y puede encriptarse mediante SSL para mayor seguridad (27).

1.6.6 Modelado

1.6.6.1 Lenguaje de modelado: UML 2.0

El Lenguaje Unificado de Modelado (UML) prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objeto y describe la semántica de lo que significan estos diagramas y símbolos. UML se usa para modelar distintos tipos de sistemas de software, de hardware y organizaciones del mundo real; ya que ofrece nueve diagramas para modelarlos (28).

1.6.6.2 Notación de modelado: BPMN

Notación para el Modelado de Proceso de Negocio es la notación gráfica que describe la lógica de los pasos en un proceso de negocio. Esta notación, se diseña especialmente para coordinar la secuencia de procesos y mensajes que fluyen entre participantes de actividades distintas. Define un modelo de procesos de negocio que se basa en diagramas de flujo.

BPMN provee una notación común para que las personas relacionadas con los procesos se expresen gráficamente de forma más clara, además estandariza y completa dicho proceso. Además, no solo estandariza los procesos dentro de la organización, sino que amplía el campo de acción para que estos puedan ser compartidos y entendidos entre los diferentes socios de negocio (29).

1.6.7 Proceso de desarrollo de software

Conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software (30).

Los procesos de desarrollo de software son aquellos que guían cómo realizar un software correctamente y tienen una implicación total en sus resultados, debido a que influyen en la calidad del mismo. Estos procesos forman un grupo de actividades a realizar y que estas, a su vez generan, un conjunto de artefactos que tributan a la correcta realización del producto.

Las metodologías ágiles o ligeras constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores que las metodologías convencionales debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración (30).

1.6.7.1 Metodología

XP (Programación Extrema)

XP es una metodología ágil para pequeños y medianos equipos, desarrollando software cuando los requerimientos son ambiguos o rápidamente cambiantes. Difiere de los procesos tradicionales desarrollados para software en que, XP asume el cambio como algo natural, y que, indefectiblemente, en alguna etapa de un proyecto sucede. Esta metodología, alienta a los programadores a responder los requerimientos cambiantes que plantea el cliente en cualquier momento, debido a que, utilizando XP se realiza el software que el cliente solicita y necesita, en el momento que lo precisa. Esto es posible porque está diseñado para adaptarse en forma inmediata a los cambios, con bajos costos asociados, en cualquier etapa del ciclo de vida. (31)

Los factores que permiten a XP ser utilizado por excelencia y brindar efectividad son, las prácticas que se realizan durante el proyecto. Entre las prácticas más utilizadas se encuentran (31):

- Cuarenta horas semanales.
- Cliente *on-site*.
- Diseño simple.

- El juego de la planificación.
- Estándares de codificación.
- Integración continua.
- Metáfora.
- Pequeñas entregas.
- Programación por pares.
- Propiedad colectiva.
- Testing.
- Refactoring.

En el desarrollo de la solución de software de la presente investigación, se utilizan algunas de las técnicas anteriormente expuestas, estas son:

- **Cliente on-site:** esta práctica, le permite al cliente la observación de la creación y avance del proyecto mientras este se construye. Esto posibilita la comunicación fluida con el cliente sin necesidad de mediar el intercambio de información a través de documentación formal. Esta comunicación constante, reduce el tiempo de desarrollo al disminuir las esperas por respuesta y tergiversaciones entre ambas partes.
- **Diseño simple:** se aplica un diseño simple en la solución de software, descartando el código complejo e innecesario, mientras que se cumpla que: corren todos los casos de prueba, se enuncian todas las ideas que se desean exhibir y no utiliza código duplicado.
- **Estándares de codificación:** se establecen reglas para que el código posea un estándar que permita consistencia, similitud y claridad.
- **Refactoring:** le permite al código la facilidad de mantención sin modificar su comportamiento. Por cuanto, se realiza permanentemente, basándose en casos de prueba; lo que posibilita que la cantidad de errores disminuya, manteniendo el diseño simple y la calidad de la solución de software.

Scrum

Proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto (32).

Las actividades que se llevan a cabo en Scrum son las siguientes:

- **Planificación de la iteración**

El primer día de la iteración se realiza la reunión de planificación de la iteración. La misma consta dos partes:

- **Selección de requisitos:** el cliente, dígame el Grupo de Servicios Legales de la Universidad de las Ciencias Informáticas, presenta al equipo de desarrollo la lista de requisitos priorizada que tiene del proceso de registro de la propiedad intelectual. El equipo de desarrollo, perteneciente al del subsistema Producción del Sistema de Gestión Universitaria, pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados en tiempo si el cliente lo solicita.
- **Planificación de la iteración:** el equipo de Producción, elabora la lista de tareas de la iteración, necesaria para desarrollar los requisitos a que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se auto-asignan las tareas.

- **Ejecución de la iteración**

Cada día el equipo realiza una reunión de sincronización. Cada miembro del equipo inspecciona el trabajo que el resto realiza, como por ejemplo: dependencias entre tareas, progreso hacia el objetivo de la iteración y obstáculos que pueden impedir este objetivo. Para de esta manera, poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido. En la reunión cada miembro del equipo responde a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener?

- **Inspección y adaptación**

El último día de la iteración se realiza la reunión de revisión de la iteración. Esta se divide en dos partes:

- **Demostración:** el equipo de desarrollo del subsistema Producción presenta al GSL los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, planifica el proyecto nuevamente.
- **Retrospectiva:** el equipo de desarrollo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad.

1.6.7.2 Proceso de desarrollo con enfoque ágil en el 2do nivel de CMMI

La Universidad de las Ciencias Informáticas lleva a cabo a partir del 2008 un proceso de mejora encaminado a alcanzar el nivel 2 del modelo CMMI. Mediante la gestión ágil de proyectos, adoptando como guía en el desarrollo del software la integración de las metodologías ágiles Scrum y Programación Extrema (XP), que tiene como objetivos dar garantías a las cuatro demandas principales de la industria en la que se ha generado: valor, reducción del tiempo de desarrollo, agilidad y fiabilidad; y CMMI, modelo de referencia para el crecimiento de capacidades y madurez, que se enfoca tanto en procesos de administración como de Ingeniería de Sistemas y Software. Para determinar la correspondencia entre la aplicación de CMMI y las metodologías ágiles XP y Scrum se realiza un análisis donde se evalúa la posibilidad de implementar las prácticas mencionadas anteriormente por el modelo y las metodologías ágiles a utilizar. Es importante destacar, que para realizar la implementación de cada práctica cumplida se propone la realización de varios artefactos a confeccionar en los proyectos de desarrollo con enfoque ágil (33).

1.7 Conclusiones

Con el estudio y análisis de los sistemas de registro de la propiedad intelectual, se logró fundamentar las bases teóricas de la investigación. Además, permitió conocer que los sistemas estudiados no cumplen los requerimientos trazados en la investigación y no es posible integrarlos al Sistema de Gestión Universitaria

porque no se desarrollan con las mismas herramientas, por lo que no son compatibles. A su vez, permitió conocer los procesos que desarrollan los sistemas y trabajos explicados, procesos que sirvieron de base para la fundamentación teórica de la presente investigación. La caracterización de las herramientas, tecnologías, metodología y lenguajes de programación permitió conocer los beneficios del uso de estas, así como formar las bases propicias para crear una propuesta de solución, que a su vez, cumpla con el objetivo de la investigación.

Capítulo 2: Propuesta de solución

2.1 Introducción

En el capítulo se realiza una descripción completa de la solución que se propone. Para esto, se hace el modelado de procesos del registro de la propiedad intelectual en la Universidad de las Ciencias Informáticas. Se especifican los requisitos funcionales a cumplir por la solución y no funcionales que presenta el sistema que se desarrollará. También se declaran los actores y sus vínculos con otros artefactos.

2.2 Modelado del negocio

El registro de la propiedad intelectual es un proceso que se inicia cuando un jefe de proyecto o autor interesado en acreditar su autoría sobre una propiedad intelectual, solicita un registro de este bien intangible al Grupo de Servicios Legales de la Universidad de las Ciencias Informáticas. Para realizar el registro se deben recoger, para cualquier tipo de obra, los datos generales que conforman la Plantilla 02.18.06.01, la cual junto con los documentos y recursos de la obra que se entregan en formato duro, ejemplos de estos son los documentos contractuales, manual de usuario, CD o DVD con copia de la obra. Luego, pasarán a ser revisados por el técnico general que se encarga de verificar que esté completa la documentación. Si es así, pasa a entregársela al especialista del grupo de propiedad intelectual que se encarga de revisar el contenido de la documentación de la obra y si no están completos notifica al jefe de proyecto para que comience el proceso nuevamente. Si está correcto el contenido de la documentación de la obra, el especialista de propiedad intelectual del GSL crea el expediente de registro, luego es el especialista el encargado de realizar la solicitud de registro en la entidad nacional, CENDA si es derecho de autor u OCPI en caso de ser propiedad industrial. Los datos son analizados por la entidad nacional responsable luego, esta entidad procede a confeccionar el certificado, luego entrega el certificado de registro al encargado del GSL, quien lo archiva y genera un nuevo certificado para entregar al autor o interesado del registro.

2.3 Modelado de procesos del negocio

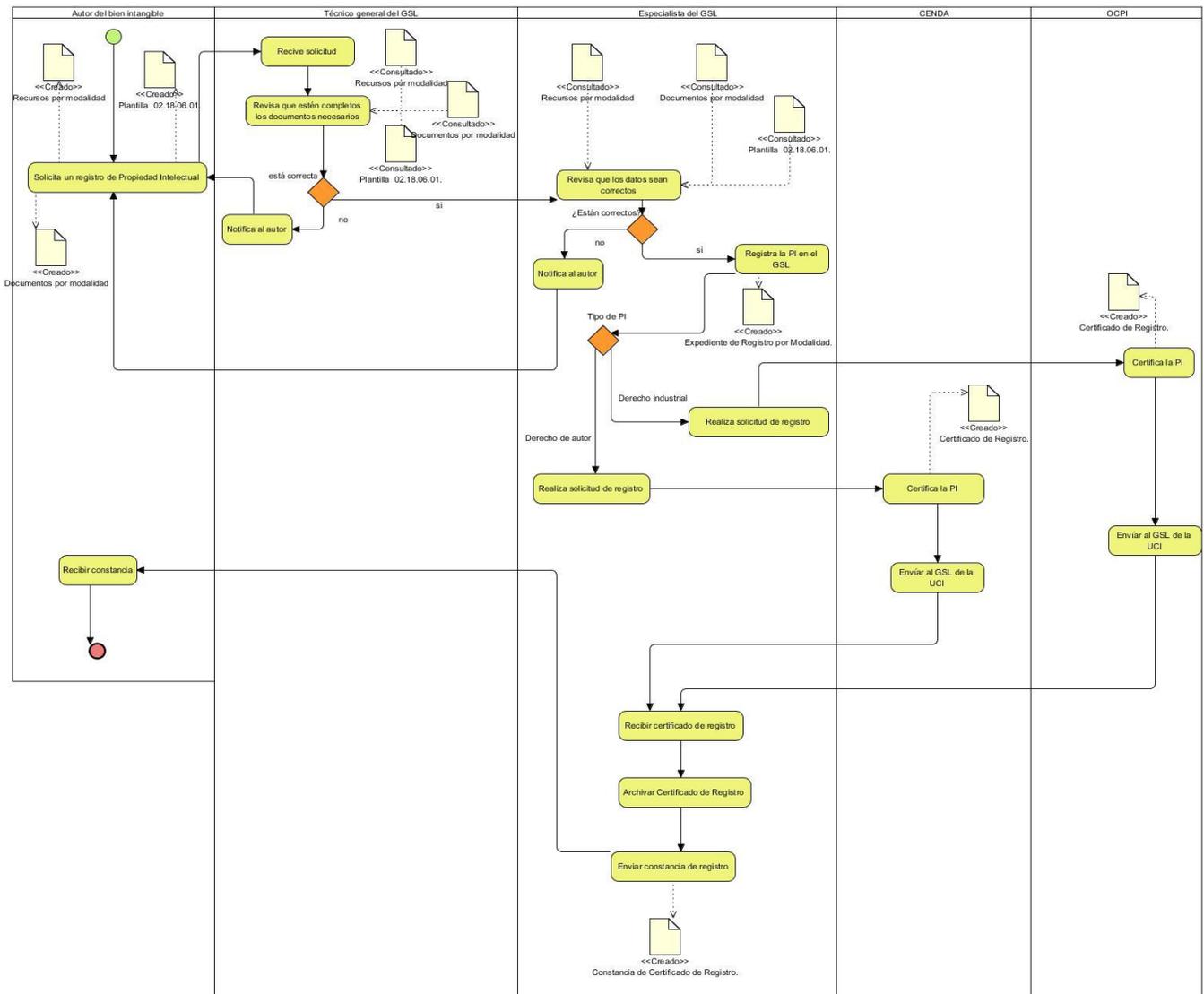


Figura 1. Modelado del proceso.

2.4 Reglas de negocio

- La solicitud de registro de una obra procede cuando su información es entregada al GSL y esta es validada.
- El registro de una obra solo solicitud de registro tiene un certificado nacional.
- Solo se otorgar el certificado de registro a una obra cuando esta es aprobada por la OCPI o por el CENDA.

- Solo el Grupo de Servicios Legales de la Dirección de la Infraestructura Productiva es el responsable en la UCI de ejecutar el proceso de registro de obras.
- Para cada proceso de solicitud de registro aprobado se genera un certificado de registro.
- Las obras solo pueden ser registradas en una de las dos instituciones nacionales encaradas de formalizar este proceso.

2.5 Propuesta del Sistema

Se propone la implementación del módulo de Registro de la propiedad intelectual (RPI) de la Universidad de las Ciencias Informáticas. El Módulo se integra al Sistema de Gestión Universitaria (SGU), se ubicada en el subsistema Producción que pertenece a dicho sistema. El módulo de RPI que se desarrolla, cuenta con cinco módulos: presupuesto, solicitud, registro, reporte y configuración. Permite generar una solicitud de registro, que el solicitante llena en consecuencia del tipo de obra que desea registrar. Debido a que los campos a llenar son en correspondencia al tipo de obra que seleccione el autor o interesado. Además permite gestionar el presupuesto para el registro de la propiedad intelectual, es decir el tipo de moneda de pago, el costo del registro y descontar del presupuesto total del GSL el costo por tipo de obra. En el Sistema se archivan los registros y se conforman reportes de estos. El tratamiento de una solicitud es según el estado en que esta se encuentra. Existen 6 estados: solicitado, denegado, revisión de documentos, registro entidad externa, registro UCI y cerrada. Solamente se permite pasar de un estado a otro si finaliza el estado actual.

La solución informática permite configurar la información que se utiliza en el proceso de registro propiedad intelectual. Con este fin, existen seis funcionalidades implementadas que permiten la gestión y manipulación de la información, según los niveles de permiso del rol autenticado. Los roles del Módulo son: administrador del sistema, técnico del GSL, especialista del GSL y jefe de centro productivo; cada uno de ellos tendrá definidos permisos tener acceso a las funcionalidades de su interés.

Con el desarrollo del módulo de RPI del SGU, se facilitan las búsquedas de información para la confección de reportes que con frecuencia son solicitados al GSL. Brinda una manera viable de realizar consultas para obtener la información que se necesita durante el proceso de registro de la propiedad intelectual. Los autores de las obras y otros involucrados en el registro de la propiedad intelectual en la UCI, pueden con el Sistema, conocer el estado de su solicitud de registro. Permite llevar el control de los períodos de vigencia de determinados registros de propiedad intelectual. Además permite la centralización efectiva de la información, agilidad en el proceso de registro de la propiedad intelectual generada en la UCI. Brinda visibilidad a la información relacionada con los registros y conocer el estado en que se encuentra el registro.

2.6 Ingeniería de requisitos

Los requisitos, reflejan las necesidades de los clientes de un sistema que ayudan a resolver un problema determinado. De igual forma Ian Sommerville describe de la siguiente manera lo que es un requisito: “los requisitos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas” (Sommerville, 2005).

2.6.1 Técnicas de obtención de requisitos

La obtención de requisitos es el proceso mediante el cual los interesados en un sistema de software descubren, revelan, articulan y entienden sus requisitos. En muchos casos, se requiere tiempo para llegar a especificar claramente lo que el interesado espera de la aplicación de software, por lo que se requiere por parte de los analistas el empleo de técnicas que permitan establecer una buena comunicación con los interesados del producto y así lograr la satisfacción del cliente. A continuación se enuncian las principales técnicas utilizadas durante el proceso de desarrollo para recopilar los requisitos de software.

- **Entrevistas:** se realizaron conversatorios abiertos y cerrados con las personas que se verían afectadas por el uso del módulo propuesto, principalmente los técnicos y especialistas del Grupo de Servicios Legales (GSL) de la Universidad de las Ciencias Informáticas (UCI), con el objetivo de entender el dominio del problema y sus necesidades.
- **Observación:** permitió comprender cómo se realiza en el GSL el proceso de registro de la propiedad intelectual, así como identificar los requerimientos implícitos en los procesos reales.
- **Prototipos:** un prototipo es una versión inicial de un sistema de software que se utiliza para demostrar los conceptos, probar las opciones de diseño y de forma general conocer más acerca del problema y sus posibles soluciones. Estos fueron mostrados al cliente, quienes proporcionaron los requerimientos adicionales.
- **Sistemas existentes:** se analizaron distintos sistemas ya desarrollados que estén relacionados con el objeto de estudio. Permitiendo la obtención de ideas para la realización del módulo a desarrollar. Muchas de las funcionalidades presentes en estos sistemas sirvieron de base para la propuesta de solución.

2.6.2 Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer (Sommerville, 2005).

A continuación se listan las principales funcionalidades que debe tener el módulo de Registro de la propiedad intelectual:

Requisitos Funcionales (RF)	Prioridad
RF1 Mostrar solicitud	Media
RF2 Ver detalles de solicitud	Alta
RF3 Crear solicitud	Alta
RF4 Mostrar revisión de documento	Alta
RF5 Ver detalles revisión de documento	Media
RF6 Crear revisión de documento	Alta
RF7 Mostrar solicitud externa	Media
RF8 Crear solicitud externa	Alta
RF9 Ver detalles de solicitud externa	Media
RF10 Modificar solicitud externa	Alta
RF11 Exportar solicitud externa	Media
RF12 Mostar presupuesto	Media
RF13 Ver detalles de presupuesto	Media
RF14 Crear presupuesto	Media
RF15 Modificar presupuesto	Media
RF16 Mostar registro	Baja
RF17 Crear notificación de registro	Media
RF18 Exportar notificación de registro	Media
RF19 Mostrar certificados	Baja
RF20 Generar reporte consumo de presupuesto por año	Media

RF21 Generar reporte cantidad de registro dado una fecha	Media
RF22 Mostrar entidad externa	Baja
RF23 Crear entidad externa	Baja
RF24 Ver detalles de entidad externa	Baja
RF25 Modificar entidad externa	Baja
RF22 Mostrar autor externo	Baja
RF23 Crear autor externa	Baja
RF24 Ver detalles de autor externa	Baja
RF25 Modificar autor externa	Baja
RF26 Mostrar tipo de moneda	Media
RF27 Crear tipo de moneda	Media
RF28 Modificar tipo de moneda	Media
RF29 Ver detalles de tipo de moneda	Media
RF30 Mostar monto por tipo de obra	Baja
RF31 Ver detalles de monto por tipo de obra	Baja
RF32 Crear monto por tipo de obra	Media
RF33 Modificar monto por tipo de obra	Media
RF34 Mostar tipo de obra	Baja
RF35 Ver detalles de tipo de obra	Baja
RF36 Crear tipo de obra	Media
RF37 Modificar tipo de obra	Media
RF38 Mostrar tipología	Baja
RF39 Ver detalles de tipología	Baja
RF40 Crear tipología	Media

RF41 Modificar tipología	Media
RF42 Mostar propiedad intelectual	Baja
RF43 Ver detalles de propiedad intelectual	Baja
RF44 Crear propiedad intelectual	Baja
RF45 Modificar propiedad intelectual	Baja

Tabla 1. Requisitos funcionales del sistema.

2.6.2 Requisitos no funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requisitos no funcionales se aplican al sistema en su totalidad (Sommerville, 2005).

A continuación se muestran los requisitos no funcionales (RNF) que debe cumplir el módulo de Registro de la propiedad intelectual:

Usabilidad	
RNF1	Se utilizará 12px como mínimo el tamaño de la fuente para lograr una mejor visibilidad de la información.
RNF2	Todas las funcionalidades estarán a nivel de dos clic como mínimo.
Seguridad	
RNF3	El acceso a la información será teniendo en cuenta los roles definidos en la aplicación para cada usuario. Se realiza actividades de chequeo ante las acciones de eliminación y modificación de la información en el Sistema, mediante mensajes de confirmación.
Legales	
RNF4	IPP-1000:2008 Elaboración y aprobación de los procedimientos y lineamientos para la actividad productiva.
RNF5	IPP-4000:2008 Registro de la propiedad intelectual.
RNF6	Resolución Rectoral No. 593/2007, Autorización al Director general de la Infraestructura Productiva a firmar la documentación que garantice los servicios y la protección de las áreas de la Infraestructura Productiva, del Rector de la Universidad de las Ciencias Informáticas
Diseño e implementación	

RNF7	IDE de desarrollo: NetBeans 7.1
RNF8	Sistema Gestor de BD: PostgreSQL 8.4
RNF9	Lenguaje de programación: PHP 5.3
RNF10	Navegador Web: Mozilla Firefox 16 o superior.
RNF11	Marco de trabajo base de desarrollo: CodeIgniter 1.7.3
RNF12	Otros componentes: JQuery 1.3.2
Software	
RNF13	PC servidor de base de datos: El servidor debe contar con sistema operativo Ubuntu Server 11.10 o CentOS 5, o una versión superior de las mismas. El SGDB debe ser PostgreSQL 8.4
RNF14	Sistema de Control de Versiones: Subversión 0.9.8
Hardware	
RNF15	Servidor de aplicaciones: 2GB RAM, 80 GB disco duro. Servidor de Base de Datos: 2GB RAM, 160 GB disco duro.
RNF16	PC cliente: 1GB RAM, 80 GB disco duro.
Apariencia o interfaz externa	
RNF17	Todas las interfaces deben cumplir las pautas de diseño definidos para el Sistema de Gestión Universitaria.
RNF18	La comunicación entre el cliente y el servidor de aplicaciones es a través del protocolo <i>Hypertext Transfer ProtocolSecure</i> (Protocolo Seguro de Transferencia de Hipertexto).

Tabla 2. Requisitos no funcionales del sistema.

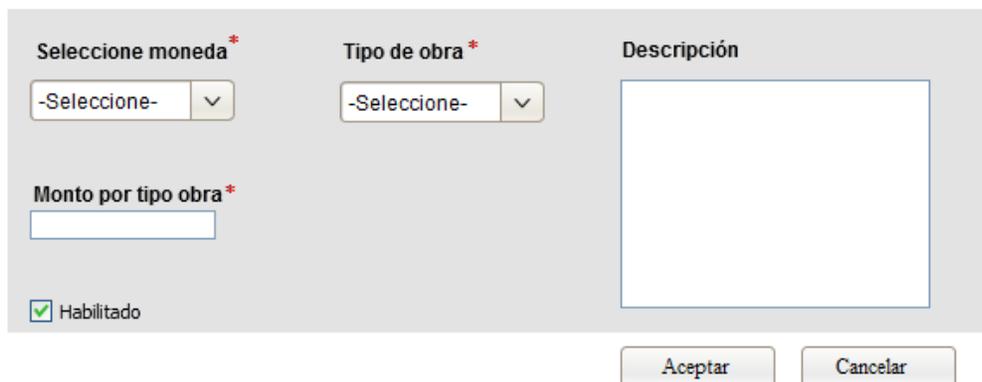
2.6.3 Especificación de requisitos

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF_33.]	Modificar monto por tipo de obra.	Permite modificar los montos por tipo de obra existentes. Se muestran los campos a modificar y además se muestra en el área de iconos flotantes las opciones de Listar y Actualizar. Una vez modificada la solicitud se actualiza el listado y se muestra un mensaje	Baja	Media

de información “El elemento ha sido modificado satisfactoriamente”.

Prototipo

Modificar monto por tipo de obra



Seleccione moneda * Tipo de obra * Descripción

-Seleccione- v -Seleccione- v

Monto por tipo obra *

Habilitado

Aceptar Cancelar

Prototipo Modificar monto por tipo de obra.

Campos	Tipos de Datos	Reglas o Restricciones
Tipo de moneda	character varying	Campo obligatorio. Selección.
Tipo de obra	character varying	Campo obligatorio. Selección.
Monto por tipo de obra	integer	Campo obligatorio. Permite la entrada de números solamente. La cantidad de caracteres permitidos para un número es de 10.

Estado(Habilitado/ Deshabilitado)	booleano	Selección.
Descripción	text	Admite a partir de 0 caracteres excepto % y comillas simples. La cantidad de caracteres permitidos para una palabra es de 30. La cantidad de caracteres permitidos es 1000.
Observaciones	<ol style="list-style-type: none"> 1. En caso que el elemento ya exista se muestra un mensaje de error "El elemento ya existe". 2. Si se modifica satisfactoriamente el elemento se mostrará un mensaje de información "El elemento ha sido modificado satisfactoriamente". 3. En caso de cancelar la acción se muestra un mensaje de advertencia "¿Está seguro de cancelar la acción?". En caso que el usuario presione el botón Aceptar se regresa al listado de empresas. Si el usuario presiona el botón Cancelar se mantiene en la misma vista. 4. En caso de que introduzca datos incorrectos (porcentaje % y comillas simples ') se muestra un mensaje de error "No se admiten apóstrofes ni porcentajes". 5. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje de error de color rojo "Campo requerido" en el campo que debe ser llenado obligatorio. 6. Si se introduce más caracteres de los permitidos para una palabra el sistema muestra el mensaje "Ha excedido el número de letras permitidas para una palabra". 7. En caso que se introduzca algún carácter que no sea número en el campo monto, el sistema muestra el mensaje de error: "Entre solo números". <p>Interactúa con esta acción el administrador.</p>	

Tabla 3. Especificación del requisito funcional 'modificar monto por tipo de obra'.

2.7 Arquitectura del sistema

La arquitectura cliente-servidor se divide en dos partes bien definidas: la primera es la parte del servidor y la segunda es un conjunto de clientes. Normalmente el servidor es una máquina suficientemente potente que puede actuar como servidor de aplicación, de depósito de datos o sencillamente brindar determinados servicios.

Por otro lado los clientes suelen ser estaciones de trabajo que realizan varias solicitudes o peticiones al servidor. Ambas partes deben estar conectadas entre sí mediante una red (34). Una representación gráfica de este tipo de arquitectura se muestra a continuación:

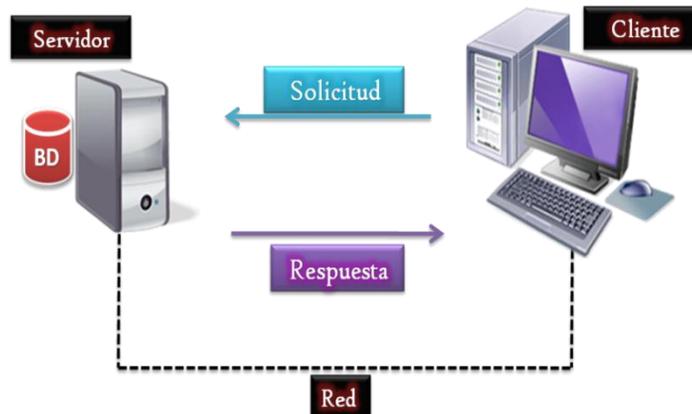


Figura 2 Arquitectura Cliente - Servidor.

2.7.1 Patrones

2.7.1.1 Patrón de la arquitectura

El patrón Modelo-Vista-Controlador (MVC) es el patrón de desarrollo en el que se basa el marco de trabajo CodeIgniter, el cual es utilizado para el desarrollo del módulo. Se ve frecuentemente en aplicaciones web, donde la vista codifica y mantiene la presentación final de nuestra aplicación de cara al usuario. En la vista se coloca todo el código HTML, CSS, JavaScript, que se tiene que generar para producir la página tal cual se quiere que la vea el usuario. El controlador es la parte más importante porque hace de enlace entre el modelo, la vista y cualquier otro recurso que se tenga que procesar en el servidor para generar la página web. En resumen, en el controlador se guarda la lógica del negocio y se realizan todas las acciones que sean necesarias para llevar a cabo los procesos del negocio. En el modelo está contenido todo el código que tiene que ver con el acceso a la base de datos, se mantiene encapsulada la complejidad de la base de datos y se crean funciones para recibir, insertar, actualizar o borrar información de las tablas. Al mantenerse todas las llamadas a la base de datos en un mismo código, desde otras partes del programa se podrán invocar las funciones que se necesitan del modelo y éste se encargará de procesarlas. En el modelo podrán preocupar cosas como el tipo de base de datos con la que se trabaja o las tablas y sus relaciones, pero desde las otras partes del programa simplemente se llaman a las funciones del modelo (35). A continuación se describe la utilización de este patrón en el desarrollo del módulo:

- **Modelo:** las clases modelos y las librerías del módulo de Registro de la propiedad intelectual (RPI), se encuentran en la capa del Modelo. Las librerías son las encargadas de organizar la lógica del negocio, mientras que las modelos son las que acceden a los datos. Las clases librerías envían los datos a las controladoras, después de obtenerlos de las modelos.
- **Controlador:** es el intermediario entre la vista y el modelo del módulo RPI. Son las clases encargadas de recibir las peticiones del usuario y en vista de satisfacer su demanda, utilizan los métodos que contienen las clases librerías para hacer peticiones de datos a las clases modelos, que son las que tienen acceso a los datos y estas al obtenerlos se los envía a través de las librerías a la controladora, que una vez que los recibe se los entrega a las vistas.
- **Vista:** son clases del módulo RPI que a través de los ficheros JavaScript, estilos CSS y la librería JQuery, conforman el código HTML necesario para mostrar la información requerida por el usuario, que obtiene de las clases controladoras.

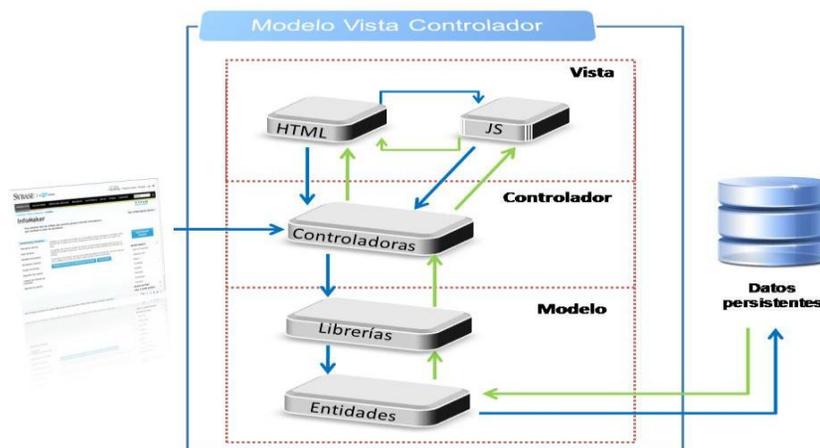


Figura 3. Patrón Modelo - Vista - Controlador.

2.7.1.2 Patrones del diseño

El uso de los patrones GRASP (*General Responsibility Assignment Software Patterns* o en español patrones generales de software para asignar responsabilidades) es fundamental en el diseño del software orientado a objeto, debido a que brindan una solución a varios de los problemas que se pueden presentar en la programación (36).

Para el diseño del módulo RPI, se tienen en cuenta los 5 patrones GRASP: experto, creador, bajo acoplamiento, alta cohesión y controlador, con el propósito de lograr un sistema reusable y flexible, debido a que estos patrones dan la medida de un refinamiento del diseño (36).

- **Experto:** este patrón se evidencia en las clases librerías del módulo que se implementa, las cuales cuentan con la información necesaria para cumplir la responsabilidad sobre los elementos del negocio. Su uso permite asignar a una clase la información necesaria para cumplir su responsabilidad, de esta forma se asignan las mismas de forma adecuada y el módulo tiende a ser más fácil de entender, mantener y ampliar, pensando en una futura reutilización de componentes.
- **Creador:** patrón que se evidencia en la clase *loader* que es el objeto load de las clases controladoras del módulo Registro de la propiedad intelectual, esta clase se encarga de cargar los elementos del marco de trabajo dígame librerías y modelos.
- **Controlador:** el patrón controlador se refleja en las clases controladoras pertenecientes al Módulo, que son las clases que se encargan de obtener datos y enviarlos a las librerías y las vistas. El patrón creador garantiza que los procesos sean manejados por la capa controladora y no por la capa de presentación.
- **Bajo acoplamiento y alta cohesión:** la propia implementación del CodeIgniter contiene estos patrones nivelados, permitiendo el uso de los componentes de forma individual en la implementación de la solución de software que se desarrolla, evidenciando de esta forma el bajo acoplamiento, así como la dependencia entre los componentes o la alta cohesión.

Además de lo anteriormente descrito, para el diseño del módulo de Registro de la propiedad intelectual se aplican de los patrones GOF (*Gang of Four*), los creacionales: fábrica abstracta e instancia única y los de comportamiento: mediador y observador. Los patrones creacionales tratan de crear instancias de objetos, el objetivo de estos es abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados. Mientras que los patrones de comportamiento ayudan a definir la comunicación e iteración entre los objetos de un sistema, con el propósito de reducir el acoplamiento entre los objetos (36).

- **Fábrica abstracta:** este patrón se refleja en el módulo Seguridad perteneciente al núcleo del Sistema de Gestión Universitaria, en la librería *fabrica_ma_lib*, que se encarga de crear los objetos de los modos de autenticación (modo de autenticación servicio web, modo de

2.9 Diagrama de despliegue

El modelo de despliegue define la arquitectura física del sistema por medio de nodos interconectados. Estos nodos son elementos de hardware sobre los cuales pueden ejecutarse los elementos de software (37).

El módulo de Registro de la propiedad intelectual, se instala en un servidor web, que se conecta con un servidor de base de datos a través del protocolo TCP/IP para la obtener los datos. La interacción al sistema se realiza mediante un navegador web de una PC cliente, se utiliza para esto el protocolo HTTPS. Los informes que se generan en el Módulo se llevan a un formato duro mediante una impresora conectada a las PC clientes.

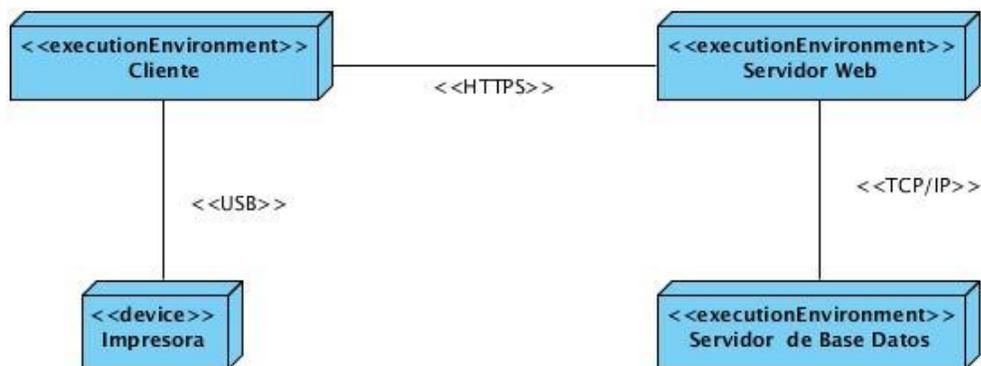


Figura 5. Diagrama de despliegue.

2.10 Conclusiones

En el presente capítulo se realizó un análisis profundo de los procesos que intervienen en las actividades de registro de la propiedad intelectual, lo que permitió la obtención del conjunto de funcionalidades que debe cumplir la aplicación y quedaron manifestadas en la especificación de requisitos del software. Además, se logró una visión clara y objetiva de los requisitos del cliente. A partir de los artefactos generados anteriormente en el capítulo, se obtuvo el análisis y diseño del módulo de Registro de la propiedad intelectual del Sistema de Gestión Universitaria, propuesta que sirve como base para las fases posteriores por las que transita la solución de software.

Capítulo 3: Implementación y pruebas del sistema.

3.1 Introducción

Entre las fases que constituyen el proceso de mejora se encuentran las de implementación y pruebas. Como resultado de la fase de implementación se desarrolla la solución propuesta, siguiendo un estándar de codificación para la implementación de las especificaciones de requisitos descritas, posibilitando la legibilidad y el mantenimiento del código en lo adelante. La realización de pruebas también adquiere alta importancia pues permite percibir la existencia de defectos en la implementación realizada.

3.2 Integración con otros módulos del Sistema de Gestión Universitaria

El subsistema Producción, que forma parte del Sistema de Gestión Universitaria (SGU), contará con el módulo de Registro de la propiedad intelectual (RPI). Este sistema se acopla al SGU, haciendo uso de las facilidades provistas para la integración, así como la reutilización de varios componentes ya implementados por otros subsistemas integrantes. Esto ha permitido la uniformidad en la arquitectura, la centralización de archivos e información y la eliminación de redundancias en el código, puesto que cualquier módulo en cualquier subsistema puede utilizar las funcionalidades y componentes implementados por los demás.

Este Sistema está integrado al Sistema de Gestión Universitaria utilizando métodos y servicios brindados por otros subsistemas y módulos dentro del mismo, entre los que se encuentran como procesos horizontales:

– Seguridad

El módulo de seguridad, se implementa a nivel central para todo el Sistema de Gestión Universitaria, garantiza el acceso a la información dados los niveles de privilegio de cada usuario, haciendo uso de la arquitectura sobre la cual está desarrollado el sistema. Esta forma de implementación de la seguridad controla no solo el acceso a un módulo determinado, sino también a cada subsistema que integra el SGU. Los roles que existen en el módulo RPI son: administrador del sistema, técnico del GSL, especialista del GSL y jefe de centro productivo.

– Configuración

La configuración del sistema, también gestionada de forma centralizada, facilita un mejor control sobre los posibles cambios que pueden ocurrir en el sistema, lo cual provee robustez y adaptabilidad al mismo. Dicho módulo permite gestionar la información de cada subsistema y sus módulos.

– Persona

El módulo Persona permite la obtención de toda la información necesaria de los posibles autores asociados a una solicitud de registro. Los datos de interés para el módulo RPI de una persona son: nombre (s), apellidos y carnet de identidad.

3.3 Integración entre módulos

Para lograr una mejor integridad de los datos, la base de datos se divide por esquemas, cada uno de estos pertenece a un módulo, de forma que cada uno de estos, solo puede acceder a las tablas que pertenecen a su esquema. Cuando se necesitan datos de otro módulo, estos se obtienen a través del IOC, archivo XML en el cual cada módulo publica métodos que devuelven datos solicitados por otros y a su vez accede a los métodos que necesita.

El módulo Registro de la propiedad intelectual se integra con el subsistema Producción, haciendo uso del módulo:

- **Proyecto:** permite conocer el rol productivo asignado a una persona vinculada a un determinado proyecto existente en la UCI.

3.4 Estándares de codificación.

Para el desarrollo del módulo se aplican los estándares de codificación establecidos para el desarrollo del Sistema de Gestión Universitaria, con el propósito de estandarizar las nomenclaturas en la implementación de la solución de software que se implementa y obtener un producto estable.

3.4.1 Identación, llaves de apertura y cierre y tamaño de las líneas

Se debe usar la indentación sin tabulaciones, con un equivalente a 4 espacios, para mantener integridad en las revisiones svn. El uso de las llaves “{}” será en una nueva línea. La longitud de las líneas de código es aproximadamente de 75-80 caracteres, para mantener la legibilidad del código.

Ejemplo:

```
public function __construct()  
{  
    parent::__construct();  
    $this->load->library("presupuesto_tipo_obra_lib");  
}
```

Figura 6. Ejemplo de indentación, llaves de apertura y cierre y tamaño de las líneas.

3.4.2 Conversión de nomenclatura

Las variables se rigen por la nomenclatura camelCase. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula.

Ejemplo:

```
$idSolicitud = $data['id'];  
$solicitud = $this->solicitud_lib->obtenerSolicitudDadoIdProducto($idSolicitud);
```

Figura 7. Ejemplo de conversión de nomenclatura.

Los nombres de las clases siempre comienzan con mayúscula, en caso de nombre compuesto las palabras se separan con el carácter subrayado “_” y el resto en minúscula.

Ejemplo:

```
class Monto_tipo_obra extends MY_Controller  
<.../>  
  
?>
```

Figura 8. Ejemplo de Conversión de nomenclatura para clases.

Las funciones se rigen por la nomenclatura camelCase. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por espacio luego de la coma que los separa.

Ejemplo:

```
function listar() {...}  
  
function listarTipoMoneda() {...}  
  
public function obtenerTipoMoneda() {...}  
  
public function registrarTipoMoneda() {...}
```

Figura 9 . Ejemplo de conversión de nomenclatura para funciones.

Los ficheros siempre se escriben en minúscula y en caso de nombres compuestos se usa el carácter subrayado”_”.

- **Vistas:** Intuitivo y relacionado con el formulario y/o vista que representa, contiene en el nombre el sufijo `_view`.
 - **Modelos:** Con el mismo nombre de la clase que representa que contiene en el nombre el sufijo `_mdl` o `_base` en caso de ser modelos base.
 - **Librerías:** Con el mismo nombre de la clase que representa que contiene en el nombre el sufijo `_lib`.
- Controladoras:** con el mismo nombre de la clase que representa.

3.4.3 Estructuras de control

Se incluye un espacio entre las estructuras de control (*if*, *for*, *foreach*, *while*, *switch*) y los paréntesis. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales. Esto aumenta la legibilidad y disminuye la probabilidad de errores lógicos.

Ejemplo:

```
if (empty($id))
{
    throw new Exception_Error('SYS007');
}
```

Figura 10. Ejemplo de estructuras de control.

3.4.4 Documentación

Todos los archivos deben de tener la documentación asociada al mismo. Para esto debe de cumplir con el siguiente bloque al principio de cada clase.

Ejemplo:

```
 * @author           Rosalina Ramírez Araluce
- */
]class Solicitud extends MY_Controller <.../>
```

Figura 11. Ejemplo de documentación.

3.4.5 Buenas prácticas

Los valores booleanos y nulos siempre se escriben con mayúscula, para facilitar la legibilidad del código, se usa una línea en blanco antes de las estructuras de control y definición de las funciones.

```
$Activo=false;
```

Figura 12. Ejemplo de buenas prácticas.

3.5 Estándares del diseño

El diseño del módulo RPI fue realizado siguiendo las pautas del diseño establecidas en el manual de directrices del Sistema de Gestión Universitaria, posibilitando la uniformidad en la estructura de todas las páginas web que componen el Módulo.

3.6 Estrategias de validación de requisitos

La validación de requerimientos trata de mostrar que estos realmente definen el sistema que el cliente desea. Esta validación es importante debido a que los errores en el documento de requerimientos pueden conducir a importantes costes al repetir el trabajo cuando son descubiertos durante el desarrollo o después de que el sistema esté en uso (38).

Para la validación de los requisitos se utilizaron las siguientes técnicas:

- **Revisiones de requerimientos:** se realizan reuniones para realizar revisiones de requisitos, ayudando a obtener los posibles errores que pueden existir en la especificación de los requisitos del software o a confirmar que los requisitos poseen la calidad deseada.
- **Generación de casos de prueba:** la elaboración de casos de prueba permite comprobar el cumplimiento de los requisitos funcionales y que estos presentan la calidad requerida.
- **Construcción de prototipos:** la confección de los prototipos permite que el usuario pueda conocer cómo es la propuesta que el equipo de desarrollo implementa y puede aprobar la idea o corregir los elementos ajenos a los requisitos funcionales especificados.

3.7 Pruebas

La prueba de software es una actividad donde un sistema o uno de sus componentes se ejecutan en circunstancias previamente especificadas, registrándose los resultados obtenidos. El objetivo de las pruebas no es asegurar la ausencia de defectos en un software, únicamente puede demostrar que existen estas anomalías. Al ejecutar las pruebas en la aplicación se utilizan técnicas experimentales, lo que permite descubrir que errores presenta. De esta manera se determina el nivel de calidad y se comprueba el grado de cumplimiento respecto de las especificaciones iniciales del sistema (38).

3.7.1 Pruebas de integración

La prueba de integración es una técnica sistemática para construir la arquitectura del software, mientras que al mismo tiempo, se aplican las pruebas para descubrir los errores asociados con la interfaz. Estas pruebas se realizan con el fin de comprobar que los módulos de un programa funcionen bien por separado y conjuntamente, debido a que un módulo puede tener un efecto adverso o inadvertido sobre otro módulo. Por tanto, el objetivo de las pruebas de integración es probar el software uniendo todos los módulos probados previamente (38).

Caso de Prueba: INT-S
Módulo al que se integra: Seguridad
Condiciones de ejecución: El módulo de Seguridad haya introducido los datos en la base de datos central y exista conexión con la misma.
Descripción de la prueba: Comprobar que el módulo “Registro de la propiedad intelectual” es capaz de realizar el acceso a funcionalidades y seguridad de negocio a partir de la información gestionada por el módulo de Seguridad.
Entradas/Pasos de ejecución: El módulo de Seguridad introduce en la base de datos central los datos y el módulo de “Registro de la propiedad intelectual” consulta estos datos y define la seguridad del módulo.
Resultado esperado: Los usuarios tienen acceso a las funcionalidades definidas de acuerdo a su rol y responsabilidades.
Evaluación: Prueba satisfactoria.

Tabla 4. Prueba de integración con el módulo Seguridad.

3.7.2 Pruebas de rendimiento y resistencia (stress)

La prueba de rendimiento está diseñada para probar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado. Esta prueba se realiza durante todos los pasos del proceso de la prueba, incluso a nivel de unidad, pues se debe asegurar el rendimiento de los módulos individuales a medida que se llevan a cabo las pruebas de caja blanca (38). Sin embargo, hasta que no estén completamente integrados todos los elementos del módulo de Registro de la propiedad intelectual, no se puede asegurar realmente su rendimiento. Las pruebas de rendimiento a menudo van emparejadas con las pruebas de resistencia y frecuentemente, requieren instrumentación tanto de software como de hardware. Las pruebas de resistencia están diseñadas para enfrentar a los programas con situaciones anormales. Esta prueba ejecuta un sistema de forma que demande recursos en cantidad, frecuencia o volúmenes

anormales. Esencialmente, el responsable de la prueba intenta romper el programa (38). Se usa la herramienta Apache JMeter 2.3 para realizar estas pruebas. JMeter es un proyecto de Apache que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones web. Los resultados de estas pruebas son los siguientes:

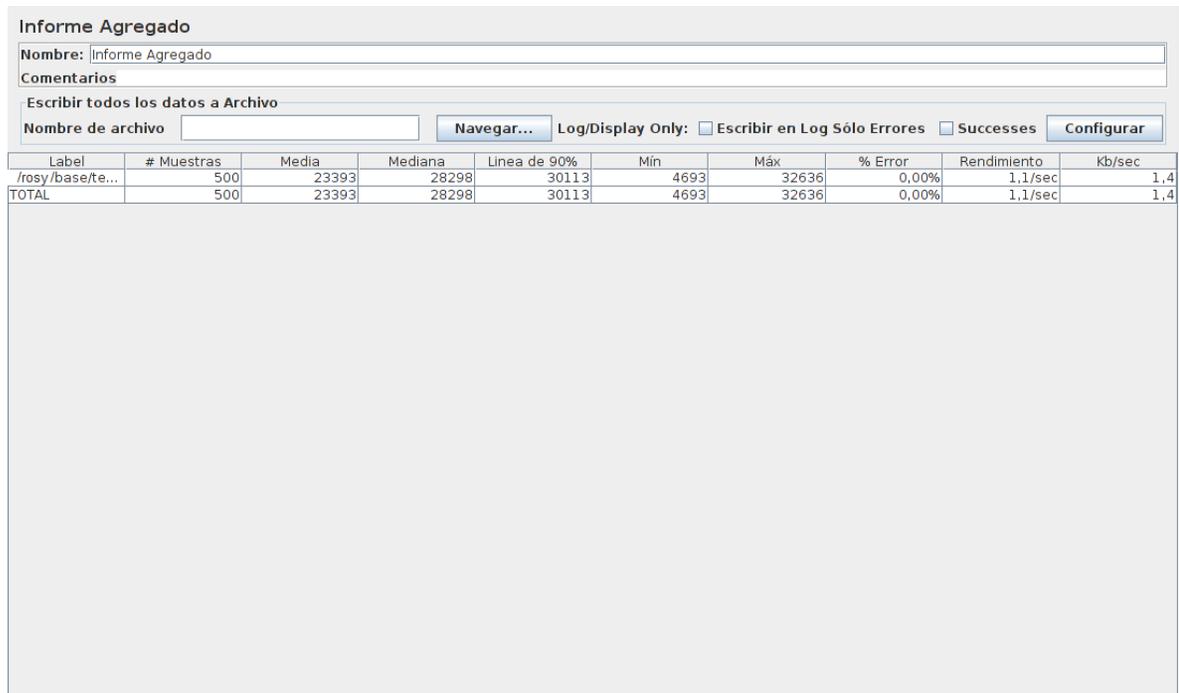


Figura 13. Prueba de resistencia para una muestra de 500 usuarios

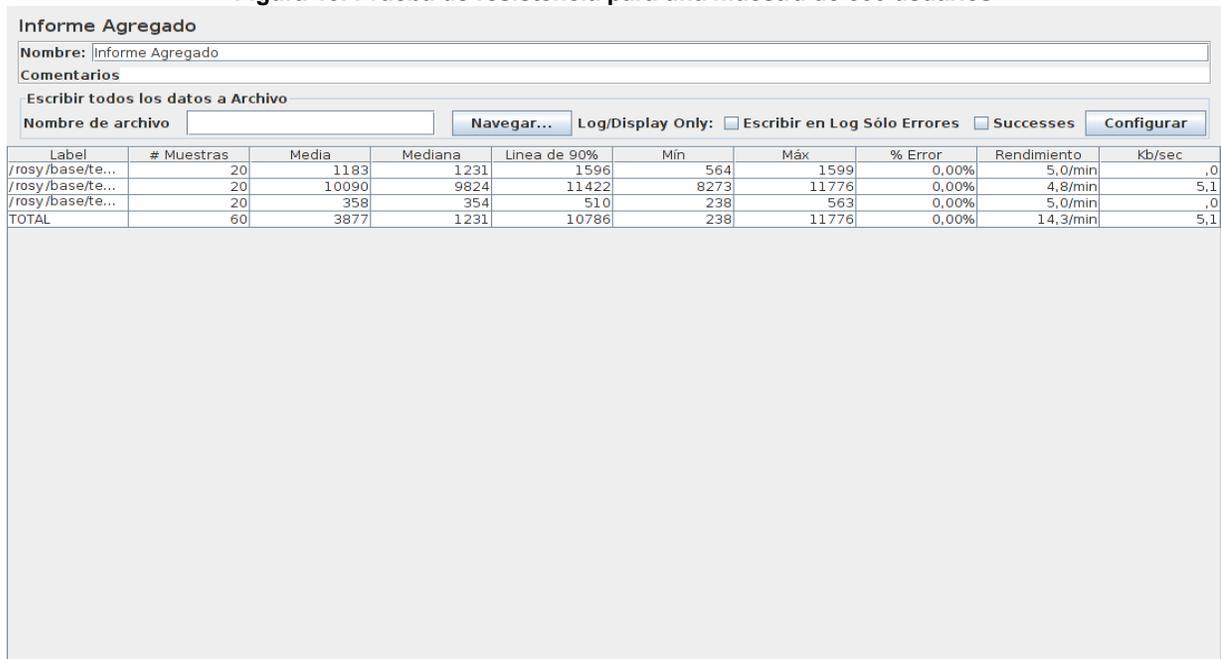


Figura 14. Prueba de resistencia para una muestra de 20 usuarios

3.7.3 Pruebas de aceptación

Las pruebas de aceptación se realizan sobre el sistema completo, no se realizan durante el desarrollo, pues sería impresentable al cliente; sino que se realizan sobre el producto terminado e integrado, o pudiera ser una versión del producto o una iteración funcional pactada previamente con el cliente. El objetivo de estas pruebas es comprobar si el cliente está satisfecho con el producto desarrollado y si este producto cumple con sus expectativas, permitiéndole al usuario que pueda determinar la aceptación del sistema (38). Esta prueba es responsabilidad del cliente, dígase el Grupo de Servicios Legales de la Universidad de las Ciencias Informáticas. Se escriben estas pruebas en colaboración con el cliente y el equipo de desarrollo del subsistema Producción del Sistema de Gestión Universitaria, para que no exista la falta de entendimiento en dicha prueba.

3.7.4 Pruebas funcionales

Se denominan pruebas funcionales a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados cumplen con los requisitos funcionales del software. A este tipo de pruebas se les denomina también pruebas de comportamiento y para realizarlas se emplea el método de caja negra, donde los probadores o analistas de pruebas no enfocan su atención en cómo se generan las respuestas del sistema, sino en el funcionamiento de la interfaz del sistema (38). Básicamente el enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida, esto generalmente se define en los casos de prueba preparados antes del inicio de las pruebas. El principal objetivo de este tipo de pruebas es demostrar que las funciones del software son operativas, que la entrada y la salida se producen de forma correcta y que además se mantiene la integridad de la información.

3.7.4.1 Casos de Prueba

Escenario	Descripción	Cantidad por página	No. de página	Respuesta del sistema	Flujo central
EC 1.1 Mostrar datos de forma correcta.	Mediante este escenario se muestra el listado de las entidades externas	NA	NA	El sistema muestra el listado de los elementos actualizados	El usuario una vez autenticado en el sistema selecciona el Subsistema de Producción y luego el módulo Registro de la propiedad intelectual. El sistema muestra las opciones del menú y el usuario selecciona en la agrupación

	existentes en el sistema. En el listado se muestra una barra de íconos internos con las acciones a desarrollar sobre cada elemento: Ver detalles y Modificar. Además se muestra una barra de íconos flotantes con las acciones a realizar sobre el contexto: Crear y Actualizar.			hasta la fecha.	funcional "Configuración" la funcionalidad "Entidad externa".
EC 1.2 Seleccionar correctamente cantidad de elementos por página.	Mediante este escenario se escoge la cantidad de elementos a mostrar por página (los valores a escoger son 5, 10, 15 y 20).	V Opción de 5.	NA	El sistema muestra la cantidad de elementos según la opción escogida (Se puede escoger 5, 10, 15, 20).	El usuario una vez autenticado en el sistema selecciona el Subsistema de Producción y luego el módulo Registro de la propiedad intelectual. El sistema muestra las opciones del menú y el usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Entidad externa".
		V Opción de 10.	NA		
		V Opción de 15.	NA		
		V Opción de 20.	NA		
EC 1.3 Escribir correctamente el número de página.	Mediante este escenario el usuario escribe debajo del listado el número de la página a la que desea acceder.	NA	V	El sistema muestra la página solicitada por el usuario.	El usuario una vez autenticado en el sistema selecciona el Subsistema de Producción y luego el módulo Registro de la propiedad intelectual. El sistema muestra las opciones del menú y el usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Entidad externa".
			2		

EC 1.4 Escribir número de página mayor que la cantidad existente.	Mediante este escenario el usuario escribe debajo del listado un número de la página a la que desea acceder, que es mayor que el total de páginas que posee el listado.	NA	120	El sistema muestra la última página del listado.	El usuario una vez autenticado en el sistema selecciona el Subsistema de Producción y luego el módulo Registro de la propiedad intelectual. El sistema muestra las opciones del menú y el usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Entidad externa".
EC 1.5 Ordenar los tipos de grupo docente correctamente.	Mediante este escenario se da clic en el encabezado del listado ordenándolo así alfabéticamente. (Ícono que se muestra como una flecha).	NA	NA	El sistema ordena los elementos de forma ascendente. Ordena primero los habilitados y luego los deshabilitados.	El usuario una vez autenticado en el sistema selecciona el Subsistema de Producción y luego el módulo Registro de la propiedad intelectual. El sistema muestra las opciones del menú y el usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Entidad externa".
				El sistema ordena los elementos de forma descendente. Ordena primero los habilitados y luego los deshabilitados.	
EC 1.6 No existen elementos creados.	Mediante este escenario en caso de que no exista creado ningún elemento se muestra un listado vacío.	NA	NA	El sistema muestra el listado vacío.	El usuario una vez autenticado en el sistema selecciona el Subsistema de Producción y luego el módulo Registro de la propiedad intelectual. El sistema muestra las opciones del menú y el usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Entidad externa".

Tabla 5. Caso de prueba para el RF_22 'mostrar entidad externa'.

3.7.4.2 Resultados de las pruebas funcionales

Al aplicar las pruebas funcionales se alcanzaron los siguientes resultados en cada una de las iteraciones aplicadas al módulo:

- En la primera iteración se analizan 18 requisitos funcionales, de los cuales se obtienen 7 no conformidades. Dentro de las cuales se puede citar: la carencia de validación en la entrada de datos en los campos obligatorios.
- Con el desarrollo de la segunda iteración se analizan 18 requisitos funcionales, quedan corregidos los errores anteriores y se detectan 4 no conformidades como es: la no visualización de los mensajes de advertencia o error. Estas no conformidades también fueron rectificadas.
- En la tercera iteración se estudian 9 requisitos funcionales. No se encontraron no conformidades, demostrando que el sistema cumple con los requisitos funcionales expuestos anteriormente.

No conformidades

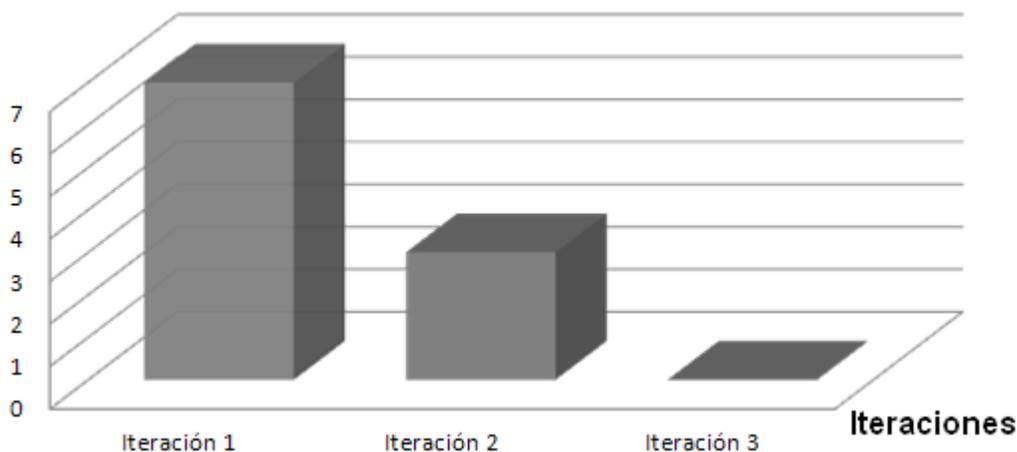


Figura 15. No conformidades del sistema.

3.8 Conclusiones

El desarrollo del presente capítulo, estrechamente relacionado con las fases de implementación y pruebas, permitió que el módulo de Registro de la propiedad intelectual quedara desarrollado y validado. La localización de las pruebas: funcionales, integración, aceptación y rendimiento y stress; arrojaron algunas deficiencias que se corrigieron y permitieron el aumento de la calidad de la solución final. De esta manera quedó confirmado que el Módulo cumple con los requerimientos definidos por el cliente.

Conclusiones generales

Al término de la presente investigación para la implementación del módulo Registro de la propiedad intelectual, integrado al Sistema de Gestión Universitaria, se obtuvieron resultados que permitieron arribar a las siguientes conclusiones:

- El estudio de los sistemas similares permitió identificar que no existen sistemas informáticos que tengan implementadas las funcionalidades que el proceso de registro de la propiedad intelectual de la UCI requiere.
- La obtención, especificación y validación de los requisitos permitió precisar las características técnicas y funcionales que definen la solución de software desarrollada así como elementos de integración con el Sistema de Gestión Universitaria.
- El desarrollo del módulo Registro de la propiedad intelectual aportó mejoras al proceso de control y registro de la propiedad intelectual de la Universidad de las Ciencias Informáticas, en cuanto a, visibilidad y mejor organización de la información.
- La aplicación de pruebas al software permitió detectar y corregir errores no identificados en la etapa de desarrollo; se logró además, la aceptación de las funcionalidades implementadas por parte del cliente.
- La solución implementada y probada permitió mejorar el proceso de registro de la propiedad intelectual de la Universidad de las Ciencias Informáticas. Disminuyó el esfuerzo físico de los trabajadores del negocio, facilitó el trabajo con la información asociada al proceso así como su consulta rápida y segura a partir de la integridad y confiabilidad de los datos. Permitted realizar el registro de bienes intangibles patrimonios de la Universidad y el manejo del presupuesto asignado al Grupo de Servicios Legales. Además posibilitó que se pueda conocer el estado actual en el que una solicitud de registro se encuentra.

Recomendaciones

Una vez concluido el objetivo de la investigación y a medida que se ha desarrollado la propuesta de solución han surgido ideas que podrían implementarse más adelante. Las funcionalidades que se recomiendan son:

- Profundizar la integración del módulo Registro de la propiedad intelectual con el módulo Proyecto, de esta forma se podrán aprovechar otras funcionalidades que este último ofrece y obtener toda la información que se necesita de un proyecto productivo, así como la relación de personas que están vinculadas a dicho proyecto.
- Realizar una integración con el módulo Notificaciones y alertas del Núcleo del Sistema de Gestión Universitaria para de esta forma mantener informado a los solicitantes del módulo Registro de la propiedad intelectual de los acontecimientos en los que se vea implicado, como por ejemplo cuando es denegada una solicitud o cuando esta es registrada en el Grupo de Servicios Legales.
- Confeccionar las ayudas contextuales para lograr un mejor entendimiento por parte del cliente sobre las funcionalidades de la aplicación.

Glosario de términos:

- **AJAX:** *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas (39).
- **CASE:** *Computer Aided Software Engineering* o Ingeniería de Software Asistida por Computadora, son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero (40).
- **Copyleft:** grupo de licencias cuyo objetivo es garantizar que cada persona que recibe una copia de una obra pueda a su vez usar, modificar y redistribuir el propio trabajo y las versiones derivadas del mismo. Unas veces se permite el uso comercial de dichos trabajos y en otras ocasiones no, dependiendo que derechos quiera ceder el autor (41).
- **Copyright:** traducido literalmente como "derecho de copia", por lo general comprende la parte patrimonial de los derechos de autor. Es un conjunto de normas jurídicas y principios que regulan los derechos morales y patrimoniales que la ley concede a los autores, por el solo hecho de la creación de una obra literaria, artística, musical, científica o didáctica, esté publicada (42).
- **CGI:** *Common Gateway Interface* (Interfaz de Entrada Común), permite a un cliente o navegador web solicitar datos de un programa ejecutado en un servidor web, especifica un estándar para transferir datos entre el cliente y el programa (43).
- **DOM:** *Document Object Model* (Modelo de Objetos del Documento o Modelo en Objetos para la Representación de Documentos), es esencialmente una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos y una interfaz estándar para acceder a ellos y manipularlos (44).
- **Form:** permite actualizar fácil y discretamente formularios HTML para usar AJAX.
- **Framework:** ambiente de trabajo y ejecución. Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.

- **FTP:** *File Transfer Protocol* (Protocolo de Transferencia de Archivos), es un protocolo de red para la transferencia de archivos basado en la arquitectura cliente-servidor.
- **GUUD:** nombre del marco de trabajo conformado con las iniciales de los departamentos del centro CENIA: Gestión Universitaria, Universidad Digital y Gestión Documental.
- **GtK:** *GIMP Tool Kit* en Inglés, es una biblioteca que contiene objetos y funciones para crear la interfaz gráficas de usuario.
- **Hash:** el término hash proviene, aparentemente, de la analogía con el significado estándar (en inglés) de dicha palabra en el mundo real: picar y mezclar. Donald Knuth cree que H. P. Luhn, empleado de IBM, fue el primero en utilizar el concepto en un memorándum fechado en enero de 1953. Su utilización masiva no fue hasta después de 10 años (45).
- **HTTP:** *Hypertext Transfer Protocol* (Protocolo de Transferencia de Hipertexto), define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor (46).
- **Knowhow:** es un término inglés que hace referencia al conjunto de conocimientos adquiridos por una empresa o un profesional, derivados de su experiencia y que pueden significar un gran valor en un mercado. El know-how o profundo conocimiento de un profesional acerca de procesos productivos, marketing, ventas u otras actividades de mercado, es un factor muy importante a la hora de definir sus honorarios (47).
- **Lenguaje interpretado:** es un lenguaje de programación que está diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes compilados.
- **MVCC:** control de concurrencia multiversión. Es un método de control de concurrencia, comúnmente utilizado por los sistemas de gestión de bases de datos para proporcionar acceso simultáneo a la base de datos y en lenguajes de programación para implementar la memoria transaccional (48).
- **Secure Hash Algorithm:** en español Algoritmo de *Hash* Seguro, es un sistema de funciones *hash* criptográficas.

- **Online:** el término en línea (*online*) hace referencia a un estado de conectividad, frente al término fuera de línea (*offline*) que indica un estado de desconexión.
- **Open Source:** equivalente en español a código abierto. Es el término con el que se conoce al software distribuido y desarrollado libremente, tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones éticas y morales las cuales destacan en el llamado software libre.
- **XML:** eXtensible Markup Language (XML) es un formato de texto simple, muy flexible derivado de SGML (ISO 8879). Originalmente diseñado para cumplir con los desafíos de la publicación electrónica a gran escala, XML también desempeña un papel cada vez más importante en el intercambio de una amplia variedad de datos en la Web y en otras partes (49).
- **XHTML:** es básicamente HTML expresado como XML válido. Es más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios o buscar errores entre otros.
- **Widgets:** es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños. Entre sus objetivos está el de brindar un fácil acceso a funciones frecuentemente usadas y proveer de información visual.

Bibliografía referenciada

1. Derecho de Autor y Propiedad Intelectual. [En línea] Universidad de Alicante . [Citado el: 15 de febrero de 2013.] <http://biblioteca.ua.es/es/propiedad-intelectual/general/derecho-de-autor-y-propiedad-intelectual.html>.
2. ECURED. [En línea] http://www.ecured.cu/index.php/Propiedad_intelectual.
3. Ortiz Sánchez, Leonides . *México y la Propiedad Intelectual*. México, D.F. : Convergencia, 2006. ISBN: 970-794-079-4.
4. Valdés Domínguez, Lic. Marta. BETSIME Revista del empresario cubano. *La Protección Jurídica de la Propiedad Intelectual*. [En línea] <http://www.betsime.disaic.cu/2004/m-j/juridmj04.htm#1>.
5. Hernández Delgado, Pedro , Navarro López, Antonio M. y Alemañy Vega, Dolores I. . *Gestión de la Propiedad Intelectual en las Empresas de Proyectos que aplican el Perfeccionamiento Empresarial*. Matanzas : Revista Avanzada Científica, 2003. ISSN 1029-3450.
6. [En línea] <http://www.uci.cu/mision> .
7. Betancourt Rodríguez, Keyly y Rodríguez Martell, Frank. *Diseño de pruebas de calidad para producto LIMS control de calidad del CIGB*. Ciudad de la Habana: Universidad de las Ciencias Informaticas : s.n., 2007.
8. Word Reference. [En línea] <http://www.wordreference.com/definicion/derecho>.
9. Reference, Word. [En línea] <http://www.wordreference.com/definicion/exclusivo>.
10. Word Reference. [En línea] <http://www.wordreference.com/definicion/titular>.
11. Creative, Safe. [En línea] <http://safecreative.org>.
12. Copyright, Semantic. [En línea] semanticcopyright.org.
13. ¿Qué es PHP? [En línea] 9 de Marzo de 2012. [Citado el: 15 de Marzo de 2012.] www.php.net/manual/es/intro-what-is.php.
14. Marley, Jimi. ¿Por qué elegir PHP? [En línea] [Citado el: 15 de Diciembre de 2011.] http://www.programacion.com/articulo/por_que_elegir_php_143.

15. Pérez, Javier Aguiluz. *Introducción a JavaScript*. 2009. pág. 5.
16. Echevarría, Álvaro Martínez. Manual Práctico de HTML. Escuela Técnica Superior de Ingenieros de Telecomunicación. Universidad Politécnica de Madrid, España. [En línea] 1995. [Citado el: 15 de Diciembre de 2011.] <http://www-app.etsit.upm.es/~alvaro/manual/manual.html>.
17. Pérez, Javier Eguíluz. *Introducción a CSS*. 2009. pág. 5.
18. —. CodeIgniter es un framework PHP para la creación rápida de aplicaciones web. Presentación general del framework y primeras notas para empezar a usarlo. [En línea] 23 de Noviembre de 2009. [Citado el: 17 de Diciembre de 2011.] <http://www.desarrolloweb.com/articulos/codeigniter.html>.
19. The jQuery Foundation. JQuery is a new kind of JavaScript Library. [En línea] 2012. [Citado el: 12 de Enero de 2012.] <http://jquery.com/>.
20. Vidal, Yanio García. *Infraestructura Productiva*. 2007.
21. PostgreSQL, El equipo de desarrollo de. *Tutorial de PostgreSQL*. [ed.] Thomas Lockhart. 1996-9. pág. 2.
22. A, Ernesto Quiñones. Introducción a PostgreSQL. [En línea] [Citado el: 15 de Enero de 2012.] http://www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf.
23. Servidor web Apache. [En línea] 8 de Mayo de 2011. [Citado el: 20 de Enero de 2012.] http://www.guia-ubuntu.org/index.php?title=Servidor_web.
24. Visual Paradigm International Ltd. Visual Paradigm for UML - UML tool for software application development. [En línea] 8 de Noviembre de 2011. [Citado el: 25 de Enero de 2012.] <http://www.visual-paradigm.com/product/?favor=vpuml>.
25. Pencil, Evolus. [En línea] <http://www.evolus.vn/Pencil/Home.html>.
26. A dedicated PHP coding environment and complete integration with web standards. [En línea] [Citado el: 29 de Enero de 2012.] <http://netbeans.org/features/php/index.html>.
27. PgAdmin III. [En línea] 10 de Marzo de 2008. [Citado el: 3 de Febrero de 2012.] http://www.guia-ubuntu.org/index.php?title=PgAdmin_III.
28. Popkin Software and Systems. *Modelado de Sistemas con UML*. pág. 2.

29. Roxana Giandini, Gabriela Pérez, Claudia Pons. *Un lenguaje de Transformación específico para Modelos de Proceso del Negocio*. LIFIA, Facultad de Informática, Universidad Nacional de La Plata. UAI, Universidad Abierta Interamericana : s.n. págs. 2-3.
30. *Experiencias del programa de mejora de procesos en la Universidad de las Ciencias Informáticas*. Kariné Ramos Blanco, Anisbert Suárez Batista, Deborat Pérez Montalván, Dennis Neuland Agüero, Ailyn Febles Estrada, Ramses Delgado Martínez, Maikel Muñoz Roja. s.l. : Ediciones Futuro, Revista Cubana de Ciencias Informáticas. ISSN: 1994-1536/RNPS: 0547.
31. Mouques, Gastón. *Metodología XP*. Uruguay : Universidad ORT Uruguay, 2003.
32. Proyectos ágiles. [En línea] <http://www.proyectosagiles.org/que-es-scrum>.
33. Castro, Ing. Eileén Llano. Propuesta para la integración de prácticas de las metodologías XP y SCRUM con el proceso de administración de requisitos del nivel 2 de CMMI.
34. Álvarez, Sara. Características principales de este tipo de arquitectura de cara a base de datos. [En línea] 30 de Agosto de 2007. [Citado el: 7 de Febrero de 2012.] <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.
35. CodeIgniter. Model-View-Controller. [En línea] [Citado el: 7 de Febrero de 2012.] http://codeigniter.com/user_guide/overview/mvc.html.
36. EVA. [En línea] eva.uci.cu.
37. Ecured. [En línea] http://www.ecured.cu/index.php/Diagrama_de_despliegue.
38. Fairley, R. *Ingeniería de software*.
39. Pergamino virtual. [En línea] <http://www.pergaminovirtual.com.ar/definicion/AJAX.html>.
40. CASE, Herramientas. [En línea] <http://fds-herramientascase.blogspot.com/>.
41. Copyleft. [En línea] <http://fundacioncopyleft.org/es/9/que-es-copyleft>.
42. Copyright. [En línea] <http://www.copyright.es/copyright.html>.
43. JavaScript, CGI y. [En línea] <http://www.masterhacks.20m.com/cgi.html>.

44. DOM Reference. [En línea] https://developer.mozilla.org/en-US/docs/DOM/DOM_Reference.
45. manual, PHP: Hash. [En línea] <http://php.net/manual/es/book.hash.php>.
46. virtual, Pergamino. [En línea] <http://www.pergaminovirtual.com.ar/definicion/Http.html>.
47. Know-how. [En línea] <http://www.headways.com.mx/glosario-mercadotecnia/definicion/know-how/>.
48. PostgreSQL. [En línea] <http://wiki.postgresql.org/wiki/MVCC>.
49. Ubiquitous Web Domain. [En línea] <http://www.w3.org/XML/>.

Bibliografía consultada

50. Sketching and Prototyping with Firefox .Pencil Project. [En línea] 2010. [Citado el: 25 de Enero de 2012.] <http://pencil.evolus.vn/en-US/Home.aspx>.

51. Definición Legal. [En línea] <http://definicionlegal.blogspot.com/2011/06/bienes-intangibles.html>.

52. Reference, Word. [En línea] <http://www.wordreference.com/definicion/registro>.

53. Pressman, R.S. *Ingeniería del Software. Un enfoque práctico.* .

54. Productiva, Infraestructura. *IPL-4020_2009 Lineamiento declaraciones legales en las creaciones UCI.*

55. Legales, Grupo de Servicios. *Registro de la Propiedad Intelectual* . 2008. IPP_4000.

56. Apache. [En línea] <http://jmeter.apache.org/>.

Anexos

Anexo 1. Especificación de requisito Mostrar monto por tipo de obra.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF_30.]	Mostrar monto por tipo de obra.	Permite mostrar un listado de los montos por tipo de obra registrados en el sistema, además muestra la opción Ver detalles en el área de iconos internos, así como las opciones Crear y Actualizar en el área de iconos flotantes.	Baja	Media

Prototipo

Monto por tipo de obra  

Filtro de búsqueda



Cantidad por página 

Monto	Tipo de obra	Tipo de moneda	
8700	Software	MN	
8700	Audiovisual	CUC	

  Página de 3   Resultados encontrados 12

Prototipo `mostrar monto por tipo de obra.`		
Campos	Tipos de Datos	Reglas o Restricciones
Observaciones	1. En caso de que no exista registrado en el sistema ningún monto por tipo de obra se muestra el listado vacío. 2. Interactúa con esta acción el administrador.	

Anexo 2. Caso de prueba SC Crear solicitud.

Escenario	Descripción	Variab le 1.	Variab le 2.	Variab le 3.	Variab le 4.	Variab le 5.	Variab le 6.	Variab le 7.	Variab le 8.	Variab le 9.	Respuesta del sistema	Flujo central
		Nombre proyecto	Nombre producto	Resumen	Estado	Área	Clasificación	Área	Clasificación	Autores asociados		
		V	V	N/A	V	V	V	V	V	V		

<p>EC 1.1 Insertar datos correctamente</p>	<p>Este escenario permite crear una solicitud insertando los datos correctamente.</p>	<p>S C A D A.</p>	<p>Com- po- nen- te Feed- back .</p>	<p>Des- crip- ción de la soli- citud.</p>	<p>H a b i l i t a d o</p>	<p>F a c i l i t a d o</p>	<p>Sis- te- ma de soft- ware</p>	<p>Soft- ware</p>	<p>AHS, ONE</p>	<p>Luis Pérez Cui.</p>	<p>El sistema muestra el mensaje: El elemento ha sido creado satisfactoriamente.</p>	<p>El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona el módulo "Registro de la propiedad intelectual" del subsistema Producción y luego en el menú lateral, en la agrupación funcional "Solicitud" selecciona la funcionalidad "Solicitud." El sistema muestra un listado con todas las solicitudes existentes hasta el momento. A continuación en el área de iconos flotantes se selecciona la opción de "Crear". El usuario introduce los datos correctamente y presiona el botón Aceptar.</p>
<p>EC 1.2 Insertar datos repetidos.</p>		<p>V</p>	<p>I</p>	<p>N/A</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>V</p>		

	<p>Mediante este escenario se pretende insertar una solicitud utilizando datos existentes en el sistema.</p>	<p>S C A D A</p>	<p>Com- po- nen- te Feed- back .</p>	<p>Des- crip- ción de la soli- citud.</p>	<p>H a b i l i t a d o.</p>	<p>F a c i l i t a d o.</p>	<p>Sis- te- ma de soft- ware .</p>	<p>Soft- ware</p>	<p>AHS, ONE</p>	<p>Luis Pérez Cui.</p>	<p>El sistema muestra el mensaje: "El elemento ya existe."</p>	<p>El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona el módulo "Seguridad" y luego en el menú lateral, en la agrupación funcional "Seguridad de negocio" selecciona la funcionalidad "Categorías de filtros." El sistema muestra un listado con todas las categorías de filtros existentes hasta el momento. A continuación en el área de iconos flotantes se selecciona la opción de "Crear". El usuario introduce los datos incorrectamente y presiona el botón Aceptar.</p>
		<p>F</p>	<p>F</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>F</p>	<p>V</p>		

EC 1.3 Insertar datos incompletos.	Mediante este escenario se dejan campos requeridos sin llenar para crear una solicitud.	(Vacio)	(Vacio)	Descripción de la solicitud.	H a b i l i t a d o.	F a c i l i t a d o. 5.	Sistema de software .	Software	AHS, ONE	Luis Pérez Cui.	El sistema muestra en rojo el mensaje: "Campo requerido" sobre el campo que debe ser llenado de forma obligatoria.	El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona el módulo "Seguridad" y luego en el menú lateral, en la agrupación funcional "Seguridad de negocio" selecciona la funcionalidad "Categorías de filtros." El sistema muestra un listado con todas las categorías de filtros existentes hasta el momento. A continuación en el área de iconos flotantes se selecciona la opción de "Crear". El usuario introduce los datos
EC 1.4 Insertar datos incorrectos.	Mediante este escenario se introducen datos incorrectos para crear una solicitud.	I	I	I	V	V	V	V	V	V		

											<i>incorrectamente y presiona el botón Aceptar.</i>
	<i>El usuario introduce menos de 2 caracteres.</i>	<i>El usuario introduce menos de 2 caracteres.</i>	<i>-</i>	<i>H a b i t a d o.</i>	<i>F a b i t a d o. 5.</i>	<i>Sistema de software.</i>	<i>Software.</i>	<i>AHS, ONE.</i>	<i>Luis Pérez Cui.</i>	<i>El sistema muestra el mensaje de error: "Entre al menos 2 caracteres" sobre el campo requerido.</i>	<i>El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona el módulo "Registro de la propiedad intelectual" del subsistema Producción y luego en el menú lateral, en la agrupación funcional "Solicitud" selecciona la funcionalidad "Solicitud." El sistema muestra un listado con todas las solicitudes existentes hasta el momento. A continuación en el área de iconos flotantes se selecciona la opción de "Crear". El usuario introduce los datos incorrectamente y</i>
	<i>I</i>	<i>I</i>	<i>N/A</i>	<i>V</i>	<i>V</i>	<i>V</i>	<i>V</i>	<i>V</i>	<i>V</i>		
	<i>"fgc '%a'"</i>	<i>"fgc '%a'"</i>	<i>Descripción de la solicitud.</i>	<i>H a b i t a d o.</i>	<i>F a b i t a d o. 5.</i>	<i>Sistema de software.</i>	<i>Software.</i>	<i>AHS, ONE.</i>	<i>Luis Pérez Cui.</i>	<i>El sistema muestra en rojo el mensaje: "No se admite apóstrofes ni caracteres especiales" sobre el campo requerido.</i>	
	<i>I</i>	<i>I</i>	<i>I</i>	<i>V</i>	<i>V</i>	<i>V</i>	<i>V</i>	<i>V</i>	<i>V</i>		
	<i>El usuario introduce más de 50 caracteres.</i>	<i>El usuario introduce más de 50 caracteres.</i>	<i>El usuario introduce más de 1000 caracteres.</i>	<i>H a b i t a d o.</i>	<i>F a b i t a d o. 5.</i>	<i>Sistema de software.</i>	<i>Software.</i>	<i>AHS, ONE.</i>	<i>Luis Pérez Cui.</i>	<i>El sistema no permite seguir escribiendo.</i>	

		te- res.	te- res.	te- res.								presiona el botón Aceptar.
		V	V	I	V	V	V	V	V	V		
		e X c r i b a.	Ges- tor do- cu- men- tal.	El u- sua- rio intro- duce más de 30 ca- rac- teres por pala- bras.	H a b i l i t a d o.	F a c u l t a d 1.	Sis- te- ma de soft- ware .	Soft- ware	AHS, ONE	Luis Pérez Cui.	El sistema muestra en rojo el mensaje:"N o se admiten más de 30 caracteres por palabras"	
EC 1.5 Cancelar operación	Me- diante este es- cena- rio se can- cela la opera- ción de crear solic- itud.	NA	NA	NA	N A	NA	NA	NA	NA	NA		
		-	-	-	-	-	-	-	-	-	El sistema muestra el mensaje de confirma- ción ¿Está seguro de realizar la acción?	El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona el módulo "Registro de la propiedad intelectual" del subsistema Producción y luego en el menú lateral, en la agrupación funcional "Solicitud" selecciona la funcionalidad "Solicitud." El sistema

muestra un listado con todas las solicitudes existentes hasta el momento. A continuación en el área de iconos flotantes se selecciona la opción de "Crear". El usuario introduce los datos incorrectamente y presiona el botón Cancelar.