

# Universidad de las Ciencias Informáticas

## Facultad 1



Título: Solución informática para gestionar la propiedad intelectual en la Universidad de las Ciencias Informáticas.

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

### **Autores:**

Carlos Yusel Carralero Calzadilla

Yusel Carlos Carralero Calzadilla

### **Tutores:**

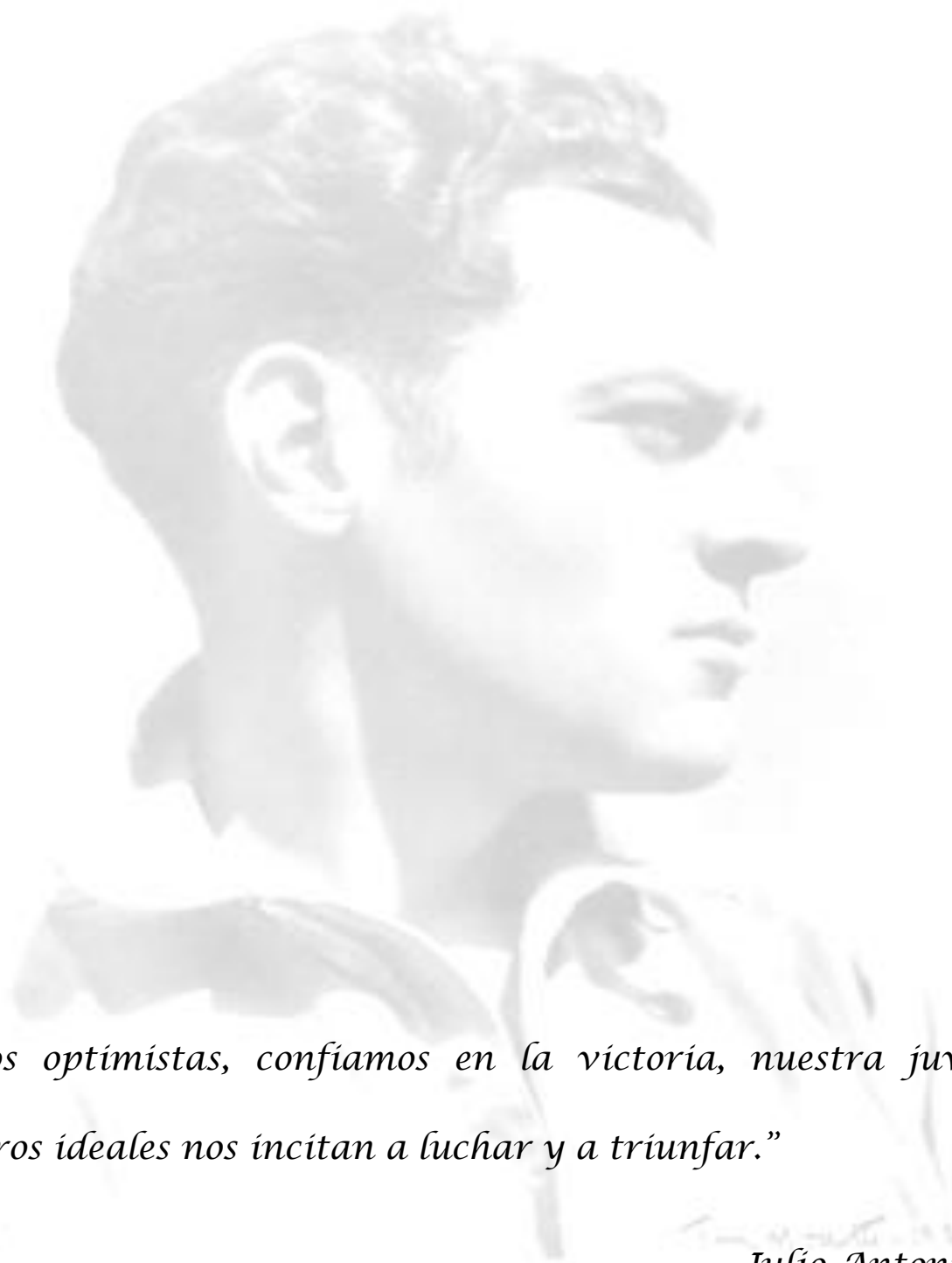
Ing. Mairelys Boeras Velázquez

Lic. Alejandro Anasagasti Cámara

Lic. Diana Aquino Ruiz

Ing. Jorge Arias Sojo

La Habana, Cuba Junio de 2014



*“Somos optimistas, confiamos en la victoria, nuestra juventud y nuestros ideales nos incitan a luchar y a triunfar.”*

*Julio Antonio Mella*

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

**Yusel Carlos Carralero Calzadilla**

\_\_\_\_\_  
Firma del Autor

**Carlos Yusel Carralero Calzadilla**

\_\_\_\_\_  
Firma del Autor

**Lic. Alejandro Anasagasti Cámara**

\_\_\_\_\_  
Firma del Tutor

**Ing. Mairelys Boeras Velázquez**

\_\_\_\_\_  
Firma del Tutor

**Ing. Jorge Arias Sojo**

\_\_\_\_\_  
Firma del Tutor

**Lic. Diana Aquino Ruiz**

\_\_\_\_\_  
Firma del Tutor

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### DATOS DE CONTACTO

#### **Tutores:**

Ing. Mairelys Boeras Velázquez: Graduada de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el 2007. Profesora con categoría de Instructor con 6 años de experiencia laboral. Se desempeñó como jefa de proyecto Pasaporte Diplomático en Venezuela. Fue asesora de mercadotecnia en el Centro de Informatización Universitaria (CENIA). Actualmente gestiona proyectos de la Dirección de Informatización (DIN) en la UCI.

Correo electrónico: mbohera@uci.cu.

Lic. Alejandro Anasagasti Cámara: Graduado de Licenciado en Matemática y Computación en el Instituto Superior Pedagógico Enrique José Varona en el 2008. Desde el 2009 trabaja en los proyectos para la construcción del Sistema de Gestión Universitaria específicamente en el sistema de Pregrado. Se desempeña como especialista de la producción.

Correo electrónico: anasagasti@uci.cu.

Ing. Jorge Arias Sojo: Graduado de Ingeniero en Ciencias Informáticas en UCI en el 2008. Profesor con categoría de Instructor con 6 años de experiencia laboral. Se desempeñó como jefe de proyecto de Intranet. Actualmente se desempeña como especialista de la DIN en la UCI.

Correo electrónico: jarias@uci.cu.

Lic. Diana Aquino Ruiz: Licenciada en Contabilidad y Finanzas, graduada en la Universidad de Oriente en el 2005. Profesora con categoría docente asistente, con 9 años de experiencia laboral. Se desempeñó como jefa de la disciplina de Ciencias Empresariales en la facultad 1, económica en uno de los proyectos de convenio realizados en Venezuela, asesora económica del Centro de Informatización Universitaria (CENIA), y actualmente es asesora de planificación y control en la DIN en la UCI.

Correo electrónico: diana@uci.cu.

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### **Dedicatoria**

*A nuestra querida madre:*

*Por darnos tanto amor y cariño desde que estuvimos por primera vez entre sus brazos, por apoyarnos siempre en las buenas y en las malas, por transmitirnos constantemente las ganas de triunfar en la vida, por el gran esfuerzo que tuvo que hacer durante estos largos años de estudio para ayudarnos a seguir adelante, por existir en nuestras vidas y ser por sí sola nuestro espíritu impulsor que nos comprende y nos da su mano para levantarnos e indicarnos el mejor camino a tomar.*

*A nuestro padre:*

*Por ser nuestro guía, nuestro mayor ejemplo ante la vida, por estar junto a nosotros en todo momento, por su preocupación constante, su atención incondicional, por enseñarnos a respetar, a ser amables, honestos, por depositar su confianza en nosotros, por brindarnos un espacio inmenso en su corazón, y sobre todas las cosas: porque siempre ha sabido llenar nuestras vidas de alegría.*

*A nuestros abuelos Berta y Ramón:*

*Por ser esas personitas adoradas que nos han acompañado desde que éramos pequeñitos, nos han reído y aplaudido nuestras travesuras, nos enseñaron las cosas buenas y malas de la vida. Por ser incondicionalmente amorosos, dedicarnos un gran tiempo de sus vidas y por siempre apoyarnos en cada cosa que necesitemos.*

*A nuestras tías:*

*Ana, Maydé, Mindá y Olguita por ser cada una de ellas un eslabón fuertísimo que aumenta nuestra confianza en nosotros mismos, por enseñarnos a cada paso que dimos una lección, por mostrarnos siempre que podemos contar con ellas para lo que necesitemos, por ser esas mujeres que jamás dudaron hasta quitarse lo que tenían para brindárnoslo y por comportarse como unas madres para nosotros.*

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### **Agradecimientos**

*A mi novia que con todo su amor y cariño a estado a mi lado apoyándome en los buenos y malos momentos, la que día a día siente mis emociones y los hace parte de ella.*

*Gracias a nuestros amigos y compañeros de la universidad en especial a Orlando, Alejandro, Yuriesky, Jose, Yasiel, Julio Cesar, Froilán, Julito, Richard, Marieli, Elaimy, Elizabet, Ismeydis y Patri por ayudarnos y apoyarnos sin interés alguno en todo lo que pudieron y por estar presentes en momentos muy importantes de nuestra vida.*

*Gracias a nuestros tutores Diana, Jorgito, Alejandro y en especial a Mairelys por su ayuda y apoyo constante, por su ardua labor y consagración en su trabajo para que lográsemos nuestros objetivos, y sobre todo por su paciencia y su poder de entendimiento ante las diferentes situaciones que se nos presentaron en el camino. A nuestros compañeros de Gestión Universitaria en especial a Mercedes, Dianela, Lía, Raidel, Guille, Alexander, Yander y Yasmany.*

*Gracias a nuestra familia por existir, por estar junto a nosotros en cada instante, por nunca abandonarnos, porque sin su apoyo jamás hubiésemos logrado este triunfo.*

# *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

## **Resumen**

En la Universidad de las Ciencias Informáticas la gestión de la información de la propiedad intelectual se torna engorrosa y difícil de manejar a partir de que la información del proceso de registro, se asienta en la herramienta ofimática Excel. La consulta de la información asociada a las solicitudes se dificulta debido a que el proceso se realiza utilizando la navegación por carpetas en la computadora donde se encuentra almacenada. La falta de visibilidad de la información, que interesa a involucrados en el proceso, afecta constantemente el trabajo en la oficina de la Dirección para la Comercialización y los Negocios (DCN) debido a las frecuentes interrupciones de estas personas solicitando información de los registros. El control del presupuesto asignado para las actividades de la propiedad intelectual se lleva en el Excel, por esta razón, se hace muy compleja la obtención de estadísticas y resúmenes de esta información.

Para dar solución a la problemática planteada, se desarrollará una solución informática orientada a facilitar la gestión de la propiedad intelectual en la UCI, sobre la base de un modelo de proceso definido, utilizando la arquitectura del Sistema de Gestión Universitaria (SGU). En el desarrollo se utiliza PHP 5.3.6 como lenguaje de programación, GUUD 1.0 como marco de trabajo, PostgreSQL 9.1 como sistema gestor de bases de datos, Apache 2.2 como servidor web, NetBeans 7.2 como entorno de desarrollo integrado y Visual Paradigm 8.0 como herramienta de modelado. Además se especifican los elementos que forman parte del diseño, arquitectura de la solución propuesta y sus funcionalidades, las cuales fueron evaluadas mediante pruebas al software.

Palabras Clave: gestión de la información, proceso de registro, propiedad intelectual.

# *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

## **Índice general**

Introducción .....	1
Capítulo 1: Fundamentos teóricos .....	6
1.1    Conceptos asociados a la problemática .....	6
1.2    Análisis de sistemas similares .....	6
1.3.1    Ámbito internacional .....	6
1.2.2    Ámbito Nacional .....	9
1.3    Herramientas a utilizar en el desarrollo de la solución .....	10
1.4.1    Herramienta para la creación de prototipos de interfaz .....	10
1.4.2    Herramienta para la realización de pruebas del sistema.....	10
1.4.3    Herramienta para el desarrollo del código fuente.....	10
1.4.4    Herramientas de modelado y diseño .....	11
1.4.5    Herramienta para la administración de la base de datos.....	11
1.4.6    Servidor web.....	11
1.4    Lenguajes y tecnologías a utilizar en la creación de la solución .....	12
1.5.1    Metodología.....	12
1.5.2    Lenguaje de modelado .....	13
1.5.3    Lenguajes utilizados .....	14
Lenguaje de acceso a datos.....	15
1.5.4    Marco de trabajo.....	15
1.5.5    Sistema gestor de base de datos.....	16
Capítulo 2: Descripción de la solución propuesta.....	18
2.1    Proceso del negocio .....	18
2.2.1    Modelado de procesos del negocio .....	19
2.2.2    Reglas de negocio .....	20
2.2.3    Base legal.....	20
2.2    Propuesta del sistema .....	21
2.3    Integración de la propuesta de solución .....	24
2.4    Ingeniería de requisitos .....	24
2.4.1    Técnicas de obtención de requisitos.....	24
2.4.2    Requisitos funcionales.....	25
2.4.3    Requisitos no funcionales .....	28



*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

2.4.4	Descripción de requisitos.....	30
2.5	Arquitectura del sistema .....	32
2.6	Patrones.....	32
2.6.1	Patrón de la arquitectura.....	32
2.6.2	Patrones del diseño .....	33
2.6.3	Patrones de base de datos .....	35
2.7	Modelo de Datos .....	36
2.8	Diagrama de despliegue.....	37
Capítulo 3: Implementación y prueba de la solución propuesta.....		39
3.1	Estándar de codificación.....	39
3.1.1	Identación, llaves de apertura y cierre, y tamaño de las líneas .....	39
3.1.2	Convención de nomenclatura .....	40
3.1.3	Estructuras de control.....	41
3.1.4	Documentación.....	41
3.2	Validación de los requisitos .....	42
3.2.1	Criterios de validación de requisitos del cliente.....	42
3.2.2	Técnicas de validación de los requisitos .....	43
3.3	Pruebas de software.....	44
3.3.1	Pruebas unitarias.....	44
3.3.2	Pruebas de integración.....	47
3.3.3	Pruebas del sistema .....	48
3.3.4	Pruebas de validación .....	52
Conclusiones generales.....		53
Recomendaciones .....		54
Glosario de términos.....		55
Referencias bibliográficas .....		56
Bibliografía.....		58
Anexos.....		62
Anexo 1 Pruebas de integración .....		62
Anexo 2 Prueba de unidad.....		63
Anexo 3 Pruebas de carga y stress.....		66
Anexo 4 Diseño de casos de prueba.....		69

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

**Índice de figuras**

Figura 1 Modelo de proceso DAC. ....	13
Figura 2 Diagrama de proceso de negocio.....	19
Figura 3 Módulo Propiedad Intelectual.....	23
Figura 4 Arquitectura cliente servidor.....	32
Figura 5 Funcionamiento del patrón Modelo-Vista-Controlador en G.U.U.D.. ....	33
Figura 6 Modelo físico de datos. ....	36
Figura 7 Diagrama de despliegue. ....	37
Figura 8 Indentación y llaves.....	39
Figura 9 Nomenclatura para crear variables. ....	40
Figura 10 Nomenclatura para crear clases.....	40
Figura 11 Nomenclatura para crear funciones. ....	40
Figura 12 Estructura de control. ....	41
Figura 13 Documentación de la clase controladora registrar.....	41
Figura 14 Prueba de carga y stress resultado para una muestra. ....	50
Figura 15 Gráfico de barra. ....	51
Figura 16 Pruebas de carga y stress resultado para 50 muestras.....	66
Figura 17 Pruebas de carga y stress resultado para 80 muestras.....	67
Figura 18 Pruebas de carga y stress resultado para 100 muestras.....	68

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

**Índice de tablas**

Tabla 1 Requisitos funcionales. ....	28
Tabla 2 Requisitos no funcionales. ....	30
Tabla 3 Descripción del requisito modificar roles productivos. ....	32
Tabla 4 Estrategia de pruebas de software. ....	44
Tabla 5 Prueba de unidad registrar entidad. ....	47
Tabla 6 Prueba de integración con el módulo Estructura y Composición. ....	47
Tabla 7 Diseño de caso de prueba listar rol. ....	49
Tabla 8 Relación de las no conformidades. ....	49
Tabla 9 Prueba de integración del módulo Propiedad Intelectual con Seguridad. ....	62
Tabla 10 Prueba de integración del módulo Propiedad Intelectual con Personal. ....	62
Tabla 11 Prueba de unidad Obtener tipo de obra. ....	64
Tabla 12 Prueba de unidad Obtener cantidad de tipos de obra. ....	65
Tabla 13 Diseño del caso de prueba crear solicitud. ....	74
Tabla 14 Diseño del caso de prueba registrar rol productivo. ....	77
Tabla 15 Diseño de caso de prueba modificar rol. ....	82

# *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

## **Introducción**

En el mundo, la creación de productos industriales, artículos, obras de artes, obras musicales, software, películas y otras creaciones del intelecto humano, han originado problemas por la forma en que se comercializaban. Muchas eran vendidas por personas ajenas a su creación y no existía la forma de verificar su autoría, lo que provocaba que terceras personas se adueñaran del producto. Con la concepción de la propiedad intelectual se resuelven en gran medida estos problemas, esta se define 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 los identificadores, los dibujos y modelos industriales y las indicaciones geográficas" (OMPI, 2008).

La propiedad intelectual a nivel mundial es regida por la Organización Mundial de la Propiedad Intelectual (OMPI). La OMPI es un organismo especializado del sistema de organizaciones de las Naciones Unidas y tiene como objetivo: desarrollar un sistema de propiedad intelectual internacional, que sea equilibrado, accesible, que recompense la creatividad, estimule la innovación, contribuya al desarrollo económico y ampare a la vez el interés público. Fue creada en 1967, con el propósito de fomentar la protección de la propiedad intelectual en todo el mundo mediante la cooperación de los estados miembros y la colaboración con otras organizaciones internacionales (OMPI, 2008).

La propiedad intelectual se divide en dos categorías: la propiedad industrial y el derecho de autor, la primera incluye las invenciones, patentes, marcas, dibujos y modelos industriales, siendo objeto de protección las ideas como soluciones a posibles problemas técnicos, contra toda utilización de su invención sin la debida autorización de su propietario, mientras que la segunda, abarca las obras literarias y artísticas, tales como: las novelas, los poemas, las obras de teatro, las películas, las obras musicales, dibujos, pinturas, fotografías, esculturas, y los diseños arquitectónicos; protege en sí la forma de expresión de las ideas, a los titulares de derechos contra todo tercero que copie o utilice la forma en que ha sido expresada la obra original (OMPI, 2008).

En Cuba la propiedad intelectual es protegida por dos organizaciones, las cuales registran y controlan todo tipo de creación propia o industrial. Una de ellas es la Oficina Cubana de la Propiedad Industrial (OCPI), órgano estatal subordinado al Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMA), responsabilizado con el establecimiento del régimen legal, la política, la gestión y la gerencia de la propiedad industrial en Cuba, así como, la prestación de los servicios inherentes a la materia (OMPI, 2008). También está el Centro Nacional de Derecho de Autor (CENDA), que fue creado el 21 de febrero de 1978 por el Decreto 20 del Comité Ejecutivo del Consejo de Ministros. El CENDA surgió por iniciativa y a propuesta del Ministerio de Cultura, se enfoca fundamentalmente en proporcionar un contexto jurídico

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

adecuado en el sistema de las relaciones que se establecen en torno a la utilización de obras intelectuales y el ejercicio de los derechos de autor (CENDA, 2010).

La sociedad cubana tiene nuevas complejidades desde el punto de vista social y económico, lo que demanda investigaciones y procesos de formación que tributen al aumento de la productividad en el trabajo. Las universidades cubanas, reúnen un alto número de profesionales que generan una elevada cantidad de obras que tributan al avance económico y social del país. La gestión y control de estas obras, que se generan en las casas de altos estudios, debe ser protegida para evitar su uso inadecuado o quebrantamiento de su originalidad. La Universidad de las Ciencias Informáticas es una muestra de estas, registrándose en el período 2010-2013 un total de 461 obras, de ellas 313 software, 12 multimedia, 34 obras científicas, 46 portales web y 22 marcas; destacándose el software y los portales web como tipos de obras. Con estos resultados se evidencia el avance de la UCI por alcanzar su misión: producir software y servicios informáticos a partir de la vinculación estudio-trabajo como modelo de formación, generando así, obras de gran valor que deben ser protegidas por las normas de la propiedad intelectual (Dirección para la Comercialización y los Negocios, 2013).

La implantación de este modelo de formación, donde se vinculan los procesos productivos, docentes e investigativos; genera un gran número de obras de alto valor para la Universidad y el país, es por ello que se ha definido un proceso mediante el cual la Universidad centralice y registre a nivel nacional la titularidad de estas creaciones del intelecto. Dentro de la Universidad, la Dirección para la Comercialización y los Negocios (DCN), es la encargada de velar por la protección de la propiedad intelectual, es por ello que ejecuta y controla todas las actividades del proceso; el cual requiere, para su correcta ejecución, la generación, actualización y consulta de un grupo de documentos que complementan el buen desempeño del mismo. Desde el año 2012 se ha estado trabajando en la reorganización y mejora del proceso y de la información asociada al mismo, pero aún no se cuenta con una solución efectiva que se ajuste a todas las necesidades. Actualmente, la información del proceso de registro, se asienta en la herramienta ofimática Excel. La información para los registros se lleva de forma centralizada en la DCN, pero no está disponible para todos los involucrados; además, su consulta se dificulta debido a que el proceso se realiza utilizando la navegación por carpetas en la computadora donde se encuentra almacenada. Para conocer el estado de una solicitud de registro de propiedad intelectual en la UCI, los autores de las obras y otros involucrados deben dirigirse a la DCN, 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 los involucrados en el proceso, y el trabajo en la oficina se ve frecuentemente interrumpido, pues las personas se presentan en la misma constantemente para conocer el estado de dichos registros. Llevar el control de los períodos de vigencia de los registros de la propiedad

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

intelectual es una tarea constante, ya que se necesita consultar frecuentemente las actas de registro para chequear que los períodos no venzan antes de ser renovados. El control del presupuesto asignado para actividades de la propiedad intelectual se lleva en Excel y obtener estadísticas y resúmenes de esta información se hace engorroso.

Teniendo en cuenta la situación problemática anteriormente planteada, se traza el siguiente **problema de la investigación**: ¿Cómo facilitar la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas?

Siendo el **objeto de estudio**: gestión de la información en el proceso de registro de la propiedad intelectual; enmarcado en el **campo de acción**: gestión de la información en el proceso de registro de la propiedad intelectual en la UCI.

El **objetivo general** que se persigue es desarrollar una solución informática orientada a facilitar la gestión de la propiedad intelectual en la UCI sobre la base de un modelo de proceso definido, utilizando la arquitectura del Sistema de Gestión Universitaria.

Para darle cumplimiento al objetivo general se definen los siguientes **objetivos específicos**:

- 1 Elaborar el marco teórico-conceptual de la investigación científica utilizando diferentes métodos para enmarcar el dominio de la misma.
- 2 Diseñar la solución informática para facilitar la gestión de la propiedad intelectual de la UCI integrada a la arquitectura del Sistema de Gestión Universitaria.
- 3 Implementar las funcionalidades diseñadas para la solución informática utilizando las tecnologías definidas.
- 4 Validar las funcionalidades de la solución de software mediante la aplicación de pruebas funcionales para garantizar la calidad del software.

Para dar cumplimiento a los objetivos específicos planteados se definen las siguientes **tareas de la investigación**:

1. Definición de los conceptos relacionados con el marco teórico de la investigación y las tecnologías necesarias para el desarrollo del sistema.
2. Análisis del estado del arte de sistemas informáticos que desarrollen procesos de la propiedad intelectual.
3. Identificación de normativas, estándares o regulaciones existentes que rijan el registro de la propiedad intelectual.
4. Especificación de requisitos funcionales y no funcionales de la solución informática.

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

5. Descripción de la arquitectura y modelado de la base de datos para el desarrollo de la solución informática.
6. Definición y descripción del uso de los patrones de diseño más adecuados para la construcción de la solución informática.
7. Diseño e implementación de la integración de la solución con módulos del Sistema de Gestión Universitaria.
8. Implementación de las funcionalidades diseñadas para la solución informática.
9. Especificación de la estrategia de prueba a seguir para el correcto desarrollo de las pruebas definidas.
10. Desarrollo y documentación de las pruebas aplicadas a la solución demostrando la calidad del software desarrollado.

Para dar cumplimiento a las tareas de la investigación planteadas se emplearon los siguientes **métodos científicos** clasificados como empíricos y teóricos según (Barchini, 2005).

### **Métodos teóricos**

**Análítico-sintético:** la utilización de este método posibilitó analizar la bibliografía relacionada con el tema. Esto permitió la extracción de los elementos más importantes que se relacionan con el objeto de estudio, pudiéndose de esta forma sintetizar el volumen de información estudiado en una nueva información concreta y resumir en ella las características generales y sus relaciones.

**Histórico-lógico:** a través de este método se adquirió el conocimiento sobre la evolución y desarrollo del registro de la propiedad intelectual. Posibilitó un mejor entendimiento del problema a resolver, detallando las necesidades que existían en la DCN de la UCI.

### **Métodos Empíricos**

**Observación bibliográfica:** este método se ha utilizado en gran parte de la investigación, pues permite el análisis de trabajos realizados con anterioridad, y la comprensión de normas y procedimientos que rigen el proceso de registro; con el fin de conocer al detalle, las particularidades del negocio, el marco legal, así como, el estado del arte en el que se enmarca la investigación para lograr un desarrollo efectivo.

**Entrevista:** este método ha sido utilizado para obtener de los expertos, información sobre la ejecución del proceso, dificultades e inconformidades que han detectado en el proceso de registro que tributen a potenciales requerimientos de software.

### **Como posibles resultados:**

- Desarrollo de la solución informática para la gestión de la propiedad intelectual en la UCI donde contribuya con la gestión de la propiedad intelectual.

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

- Mejor control del presupuesto asignado a la DCN para las actividades de propiedad intelectual.
- Incorporación de funcionalidades en la solución de software que flexibilizan su configuración y que le permitan su adaptación fácil a los cambios de los procesos de negocio: gestión de áreas de registro, gestión de entidades titulares de obras, gestión de documentos para solicitudes, gestión de roles para autores, gestión de actividades y asociar áreas para registro a usuarios.
- Mejor control de las entidades titulares de obras con la Universidad.
- Aumento del control de los períodos de vigencia de registros de propiedad intelectual para las obras que lo requieran.
- Mayor visibilidad de la información que relaciona el proceso de registro, permitiendo a las áreas de la UCI interesadas saber el estado de registro de sus obras en cada momento.
- Obtención de reportes relacionados con el proceso, el presupuesto asignado para el registro y con las obras.
- Disminución de tiempo en búsquedas, consolidación de información, confección y ajuste de formatos para documentos.

Este trabajo está estructurado por tres capítulos. A continuación se resume el contenido de los mismos:

- **Capítulo 1: Fundamentos teóricos:** se definen los principales conceptos asociados al dominio de la problemática existente. Se realiza una investigación sobre los sistemas que gestionan la propiedad intelectual. Se describen las principales herramientas, tecnologías y lenguajes seleccionados por la Dirección de Informatización (DIN) para el desarrollo de la solución informática que da nombre a esta investigación.
- **Capítulo 2: Descripción de la propuesta de solución:** se realiza un estudio del proceso de negocio y se describe la propuesta de solución enmarcada en la gestión de la propiedad intelectual existente en la universidad. Se describen los requisitos funcionales y no funcionales con los que debe cumplir la propuesta de solución, así como, los elementos del diseño y la arquitectura. Además, se presenta el modelo de datos y el diagrama de despliegue de dicha propuesta.
- **Capítulo 3: Implementación y validación de la solución:** en este capítulo se describe la implementación de la propuesta de solución teniendo en cuenta los estándares de programación y las técnicas de codificación. Además, se presentan las pruebas que servirán para validar la solución como las pruebas unitarias, de integración, del sistema (rendimiento y resistencia) y de aceptación para satisfacer todas las necesidades del cliente.



# *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

## **Capítulo 1: Fundamentos teóricos**

### **Introducción**

En el presente capítulo se definen los principales conceptos asociados al dominio de la problemática existente. Se valoran los sistemas existentes que manejan la gestión de la propiedad intelectual con el objetivo de estudiar sus características y funcionalidades. Se realiza un resumen acerca de las principales herramientas, tecnologías y lenguajes seleccionados para el desarrollo de la solución ajustándose a los requerimientos del Sistema de Gestión Universitaria.

### **1.1 Conceptos asociados a la problemática**

- **Bien intangible:** aquellos bienes que no poseen materialidad, pero sí aportan valor y suponen un valor añadido. Ejemplos son: las marcas, diseños industriales, obras de artes, por citar algunos.
- **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 UCI, las obras susceptibles a registro son: audiovisuales, marcas, software, base de datos, obras de arte, científicas, diseño de comunicación social, fotográfica, literaria y auditorías DVD.
- **Entidad cotitular:** entidad que interviene en la creación de la obra.
- **Entidad externa:** entidad que rige, controla y registra las obras con el fin de darle protección o lograr objetivos gubernamentales.
- **Autor externo:** persona que se considera autor de la obra, pero que no pertenece a la UCI.

### **1.2 Análisis de sistemas similares**

#### **1.3.1 Ámbito internacional**

##### ***Plataforma Web para la Gestión de Patentes***

Esta plataforma gestiona los derechos de propiedad industrial asociados a los resultados de los programas y proyectos de investigación que se ejecutan en la Universidad de Talca, Chile. El trabajo se inició con una investigación exploratoria seguida de la construcción de un modelo de procesos haciendo uso de la Notación para el Modelado de Procesos de Negocio (BPMN por sus siglas en inglés). Finalmente, se procedió a desarrollar la plataforma web aplicando la metodología de desarrollo ágil de proyectos de Programación Extrema (XP por sus siglas en inglés). El resultado alcanzado de la plataforma web, ha posibilitado la formalización de procesos, la definición de reglas, la identificación de roles por

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

parte de los distintos actores involucrados, y está facilitando el monitoreo de las solicitudes de protección de los derechos de propiedad industrial en la Universidad. El sistema permite: manejar información consolidada sobre la gestión del proceso de patentes; desplegar informes y reportes que serán fundamentales para las mejoras propias del proceso; y apoyar la toma de decisiones en los casos que sea requerido (Acuña, y otros, 2011).

Descrita la plataforma web se concluye que este sistema gestiona solamente el proceso de las patentes en la ciudad de Chile bajo las normas de ese país. Además, fue desarrollado mediante el uso de tecnología Asp.Net, el cual es un marco de trabajo para aplicaciones web y utiliza el entorno de desarrollo de Visual Studio.Net bajo el lenguaje Visual Basic.net. Esta plataforma no satisface las necesidades de la investigación ya que utiliza tecnologías propietarias y no cumple con las políticas cubanas del uso de software libre en el desarrollo de soluciones informáticas.

### **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, proporciona información, seguridad y autogestión de los derechos de propiedad intelectual (SafeCreative, 2008).

- 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* (SafeCreative, 2008).

- 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 auto distribución 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 (SafeCreative, 2008).

- Descripción de la plataforma

*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

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

misma por medio de licencias/contratos predefinidos como: *Creative Commons*, Licencia Pública General (LPG) u otros sistemas.

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, sellado de tiempo redundante de la fecha y hora del registro.
- Interfaz abierta para registro y consulta desde sistemas externos.
- Marco tecnológico empleado: *Semantic Copyright*.
- Contratación de licenciamiento directo con el autor.
- Registro gratuito (*SafeCreative*, 2008).

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 en 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 las obras. 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 informático 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 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* y el departamento de Inteligencia Artificial de la Universidad Politécnica de Madrid con el apoyo del Ministerio de Industria, Turismo y Comercio de España (*Semantic Copyright*, 2009).

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:

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

- **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.
- **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 (Semantic Copyright, 2009).

Por todo lo anteriormente planteado, se llega a la conclusión de que *Semantic Copyright* es un sistema informático que permite identificar si una obra es una copia de la original por su morfología. Luego comprueba la morfología del fichero seleccionado, en el subsistema identificación de la obra, perteneciente al marco tecnológico, a través de huellas digitales. Por lo tanto, ya que se desea gestionar 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.

### **1.2.2 Ámbito Nacional**

Como parte de la investigación, se realizó una búsqueda en Cuba de sistemas que relacionen el negocio tratado en la presente investigación, arrojando como resultado la existencia de un trabajo de diploma en la UCI durante el curso escolar 2012-2013, que trata en cierta medida la problemática abordada en este trabajo. Esta solución informática para el control del registro de la propiedad intelectual en la Universidad de las Ciencias Informáticas fue implementada en el SGU, el cual dio los primeros pasos en función de resolver los problemas con la gestión de la propiedad intelectual en la UCI. Esta solución no incluye la totalidad de los procesos que la DCN requiere, el mismo no llegó a implantarse debido a problemas en la implementación y por la ausencia de importantes procesos como es la adaptabilidad a posibles cambios en el proceso de registro, la falta de control de los estados por los que transita la solicitud de registro y la

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

falta de control del presupuesto destinado a las actividades vinculadas a cada registro. A pesar de los problemas detectados constituye un importante pilar en el desarrollo del presente trabajo.

Los sistemas estudiados anteriormente, aunque permiten de una forma u otra gestionar la propiedad intelectual, presentan inconvenientes, las cuales ya fueron definidas, por tal razón, surge la solución informática para facilitar la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

### **1.3 Herramientas a utilizar en el desarrollo de la solución**

Para el desarrollo de la solución se utilizarán las herramientas definidas por el grupo de arquitectura, tecnología y soporte de la DIN.

#### **1.4.1 Herramienta para la creación de prototipos de interfaz**

##### **Evolus Pencil 2.0.5**

Es una herramienta gratuita y de código abierto para hacer diagramas y prototipos. Las principales características son: puede ser instalado como extensión para Firefox o como una aplicación independiente para Windows, GNU / Linux y MacOS X. Permite la exportación al Lenguaje de Marcas de Hipertexto (HTML por sus siglas en inglés), PNG, documento de Word y PDF. Además, se pueden realizar operaciones de dibujo estándar: alineación, z - orden, escalado, rotación entre otras. También se le pueden adicionar objetos externos. Por tal razón, se decide utilizar la herramienta para el desarrollo de los prototipos de interfaz de la propuesta de solución (Pencil project, 2012).

#### **1.4.2 Herramienta para la realización de pruebas del sistema**

##### **JMeter 2.3.4**

La aplicación de escritorio Apache JMeter es un software de código abierto, una aplicación de Java, diseñado para cargar el comportamiento funcional de prueba y medir el rendimiento. Originalmente fue diseñado para aplicaciones de pruebas Web, pero desde entonces se ha expandido a otras funciones de prueba (Apache Software Foundation, 2013).

#### **1.4.3 Herramienta para el desarrollo del código fuente**

##### **Entorno de Desarrollo Integrado**

El Entorno de Desarrollo Integrado (IDE por sus siglas en inglés) es un programa de aplicación que consiste en un editor de código, compilador, depurador y constructor de interfaz gráfica. Pueden ser

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación.

### **Netbeans 7.2**

*NetBeans IDE* es una aplicación de código abierto, diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas. Hace uso de la tecnología Java y dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas y creación de aplicaciones compatibles con teléfonos móviles, además, se pueden realizar todas las tareas asociadas con la programación como editar código, compilarlo, ejecutarlo y depurarlo. También posee una vibrante comunidad de desarrolladores, ofrece una amplia documentación y tiene soporte para el Protocolo de Transferencia de Archivo (FTP por sus siglas en inglés) y sistema de control de versiones (Oracle, 2013).

### **1.4.4 Herramientas de modelado y diseño**

#### **Visual Paradigm para UML 8.0**

Visual Paradigm para Lenguaje Unificado de Modelado (UML por sus siglas en inglés) es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software ayuda a la construcción de aplicaciones con mejor calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Además, es multiplataforma y cuenta con una versión libre para la comunidad (Visual Paradigm para UML, 2010).

### **1.4.5 Herramienta para la administración de la base de datos**

#### **Administrador de base de datos PgAdmin III 1.12.1**

Es una aplicación gráfica para gestionar el gestor de bases de datos *PostgreSQL*, siendo la más completa y popular con licencia de código abierto. Está escrita en el lenguaje de programación C++ usando la librería gráfica multiplataforma, lo que permite que se pueda usar en Linux. Es capaz de gestionar versiones a partir de la *PostgreSQL* 7.3 ejecutándose en cualquier plataforma, así como, versiones comerciales de *PostgreSQL*. *PgAdmin III* está diseñado para responder 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 (pgAdmin, 2010).

### **1.4.6 Servidor web**

#### **Apache 2.2**

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

Apache está diseñado para ser un servidor web potente y flexible. Puede funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y los diferentes entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades, o que una misma característica o funcionalidad sea implementada de diferente manera para obtener una mayor eficiencia. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios web elegir qué características van a ser incluidas en el servidor seleccionando y qué módulos se van a cargar, ya sea al compilar o al ejecutar el servidor (Apache, 2012).

### **1.4 Lenguajes y tecnologías a utilizar en la creación de la solución**

#### **1.5.1 Metodología**

##### **Desarrollo Ágil con Calidad (DAC)**

La metodología para el Desarrollo Ágil con Calidad (DAC) es un proceso de desarrollo de software que combina las metas y prácticas de las áreas de procesos del nivel 2 de Integración de Modelos de Madurez de Capacidades (CMMI por sus siglas en inglés) con las buenas prácticas de la dirección y desarrollo ágil de proyectos de software. Es un proceso colaborativo, recursivo-iterativo, incremental y guiado por procesos y requisitos. Es una adaptación del modelo en Cascada a los modelos Programación extrema y Desarrollo concurrente. Está enfocado a proyectos pequeños o proyectos grandes divididos en sub-proyectos que desarrollan software de gestión basado en componentes.

Este proceso tiene 8 actividades del marco de trabajo del proceso común, llamadas Fases o Procesos del ciclo de vida: Inicio, Análisis y Diseño arquitectónico, Requisitos, Construcción, Cierre de iteración (opcional), Liberación, Transición y Cierre, ocurriendo las iteraciones concurrentes entre las fases de Requisitos, Construcción y Cierre de iteración. Además, entre las fases de Requisitos y Construcción puede ocurrir un ciclo pues a medida que los requisitos son especificados estos pueden ir entrando a la fase de Construcción. El proceso tiene también dos áreas de procesos de protección: Gestión de proyectos y Soporte, así como, dos Fases o Procesos Horizontales cuyas tareas están presentes en varias de las fases del proceso común en forma de subprocesos: Arquitectura y Planificación (Sánchez Méndez, 2013).

##### **Integración de Modelos de Madurez de Capacidades (CMMI) Nivel 2**

El CMMI es un modelo de calidad del software que clasifica las empresas en niveles de madurez. Estos niveles sirven para conocer la madurez de los procesos que se realizan para producir software. La representación por etapas ofrece una manera sistemática y estructurada de aproximarse a la mejora de procesos basada en el modelo etapa a etapa. En el nivel de madurez dos, los proyectos de la organización han asegurado que los procesos se planifican y realizan de acuerdo a las políticas, los



## Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

proyectos emplean personal con habilidad que dispone de recursos adecuados para producir resultados controlados, involucran a las partes interesadas relevantes, se monitorizan, controlan, revisan y se evalúan en cuanto a su adherencia a sus descripciones del proceso.

### Enfoque Ágil

El desarrollo ágil de software son métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan mediante la colaboración de grupos organizados y multidisciplinarios. Existen muchos métodos de desarrollo ágil; la mayoría minimiza riesgos desarrollando software en lapsos cortos. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requisitos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, sino que la meta es tener una versión sin errores al final de cada iteración. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto (Canós, y otros, 2008).

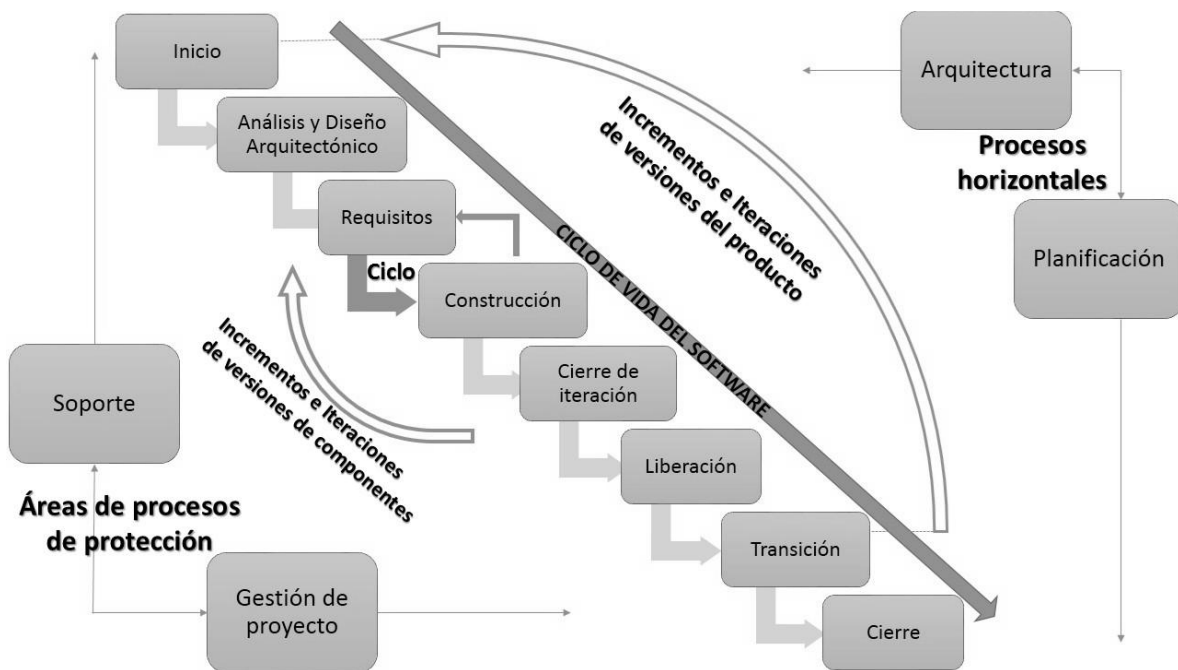


Figura 1 Modelo de proceso DAC.

### 1.5.2 Lenguaje de modelado

#### UML

El Lenguaje Unificado de Modelado (UML por sus siglas en inglés) es un popular lenguaje de modelado de sistemas de software. Se trata de un lenguaje gráfico para construir, documentar, visualizar y especificar



## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

un sistema de software. Entre otras palabras, UML se utiliza para definir un sistema de software. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medio (ALEGSA, 2010).

### **1.5.3 Lenguajes utilizados**

#### **HTML 4**

El Lenguaje de Marcas de Hipertexto (HTML por sus siglas en inglés) se basa en el metalenguaje y es el formato de los documentos en la Red de Amplitud Mundial (WWW por sus siglas en inglés). HTML se encarga de desarrollar una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementando dicho texto con diversos objetos como fotografías y animaciones. El texto en él se crea a partir de etiquetas, también llamadas *tags*, que permiten interconectar diversos conceptos y formatos. Para la escritura de este lenguaje, se crean etiquetas que aparecen especificadas a través de corchetes o paréntesis angulares: < y >. Entre sus componentes, los elementos dan forma a la estructura esencial del lenguaje, ya que tienen dos propiedades, el contenido en sí mismo y sus atributos. Por otra parte, cabe destacar que el HTML permite ciertos códigos que se conocen como *scripts*, los cuales brindan instrucciones específicas a los navegadores que se encargan de procesar el lenguaje. Entre los scripts que pueden agregarse, los más conocidos y utilizados son *JavaScript* y PHP (aprenderaprogramar.com, 2013).

#### **JavaScript 1.8**

*JavaScript* es un lenguaje de programación interpretado, o sea, no requiere compilación. Es utilizado especialmente en páginas web embebido en el código HTML o similares. La mayoría de los navegadores pueden interpretar los códigos *JavaScript* incluidos en las páginas web. Además, es un lenguaje basado en prototipos, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros (Mozilla Developer Network, 2013).

#### **PHP 5.3.6**

El Preprocesador de Hipertexto (PHP por sus siglas en inglés) es un lenguaje de programación usado generalmente en la creación de contenidos para sitios web. Es un lenguaje interpretado especialmente usado para crear contenido dinámico web y aplicaciones para servidores, aunque también es posible crear aplicaciones gráficas. La interpretación y ejecución de los scripts PHP se hacen en el servidor, el cliente solo recibe el resultado de la ejecución y jamás ve el código PHP. También permite la conexión a todo tipo de servidores de base de datos (Softonic Internacional S.A., 2012).

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### **Lenguaje de acceso a datos**

El Lenguaje de Consultas Estructurado (SQL por sus siglas en inglés) es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones sobre los mismos. Es un lenguaje declarativo de alto nivel o de no procedimiento, permite una alta productividad en codificación (Microsoft, 2014).

El SQL proporciona una rica funcionalidad más allá de la simple consulta o recuperación de datos. Asume el papel de lenguaje de definición de datos, lenguaje de definición de vistas y lenguaje de manipulación de datos. Además, permite la concesión y denegación de permisos, la implementación de restricciones, integridad y controles de transacción (Microsoft, 2014).

### **1.5.4 Marco de trabajo**

#### **JQuery 1.7.1**

Es un marco de trabajo en *JavaScript* para facilitar el acceso a los elementos del Modelo de Objetos del Documento (DOM por sus siglas en inglés), los efectos, interactuar con los documentos HTML, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web. La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX. Entre las principales características se tiene la interactividad y modificaciones del árbol DOM, el manejo de eventos, la manipulación de la Hoja de Estilos en Cascada (CSS por sus siglas en inglés), el manejo de efectos y animaciones y el soporte de extensiones (jQuery Foundation, 2013).

#### **CodeIgniter 1.7.3**

Es un marco de trabajo que utiliza el lenguaje PHP para la creación de cualquier tipo de aplicación web de código libre. Contiene una serie de librerías que sirven para el desarrollo de aplicaciones web, además, propone una manera de desarrollarlas que se debe seguir para obtener provecho de la aplicación. Marca una manera específica de codificar las páginas web y clasificar sus diferentes *scripts*, que sirve para que el código esté organizado y sea más fácil de crear y mantener (Scribd, 2013).

Entre sus características se encuentran:

- Sistema basado en Modelo-Vista-Controlador.
- Extremadamente liviano.
- Soporte de active record para base de datos.
- Formulario y validación de datos.
- Manejo de sesión.

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

- Clase de envío de email. Soporta archivos adjuntos, email de texto/HTML y múltiples protocolos.
- Librería de manipulación de imagen (cortar, redimensionar, rotar, etc.).
- Clase de carga (upload) de archivo.
- Clase de FTP.
- Localización.
- Paginación.
- Encriptación de Datos.
- Cacheo de páginas enteras.
- Historial de errores.
- Clase de Calendario.
- Clase de Codificación Zip.
- Clase de Prueba de Unidad.
- URLs amigables a motores de búsqueda.
- Soporte para ganchos, extensiones de clase y plugins.
- Larga librería de funciones “asistentes” (Álvarez, 1999).

### **Marco de trabajo G.U.U.D.**

Es un marco de trabajo desarrollado por el grupo de arquitectura de la DIN para la creación de aplicaciones web escritas en PHP. Constituye un híbrido entre el marco de trabajo *CodeIgniter* 1.7.3 y *jQuery* 1.7.1 en una sola infraestructura. Implementa el patrón Modelo-Vista-Controlador (MVC por sus siglas en inglés), así como, la programación orientada a aspectos. Realiza un manejo de excepciones y mensajes. Modulariza el marco de trabajo *CodeIgniter* e implementa como estrategia de comunicación entre módulos la Inversión de Control (IoC por sus siglas en inglés). Hace uso de templates para el rederizado de las vistas. Contiene una serie de componentes visuales que permiten la fácil interacción del usuario con la aplicación tales como: el calendario y el *grid*, entre otros (Rodríguez Mompié, y otros, 2013).

### **1.5.5 Sistema gestor de base de datos**

#### **PostgreSQL 9.1**

Es un sistema de gestión de bases de datos objeto-relacional ya que incluye características de la orientación a objetos, como pueden ser: la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos (PostgreSQL, 2009).

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

Principales características de este gestor de bases de datos:

- ✓ Implementa el estándar SQL92/SQL99.
- ✓ Soporta distintos tipos de datos; además, del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP y otras), cadenas de bits, etc.
- ✓ Incorpora una estructura de datos array.
- ✓ Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- ✓ Permite la declaración de funciones propias, así como la definición de disparadores.
- ✓ Soporta el uso de índices, reglas y vistas.
- ✓ Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- ✓ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos (PostgreSQL, 2009).

### **Conclusiones parciales**

Con la realización de esta primera parte de la investigación se definieron los conceptos relacionados con el proceso lo que permitió un mejor entendimiento entre el cliente y los desarrolladores del sistema. Además, al analizar los sistemas informáticos, se logró evidenciar la necesidad de desarrollar una solución informática que se ajuste a las características y requerimientos del proceso de registro que lleva a cabo la DCN en la UCI. La caracterización de los lenguajes y herramientas que se utilizarán en el trabajo, posibilitó conocer sus particularidades con el fin de hacer un mejor uso de sus bondades.

# *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

## **Capítulo 2: Descripción de la solución propuesta**

### **Introducción**

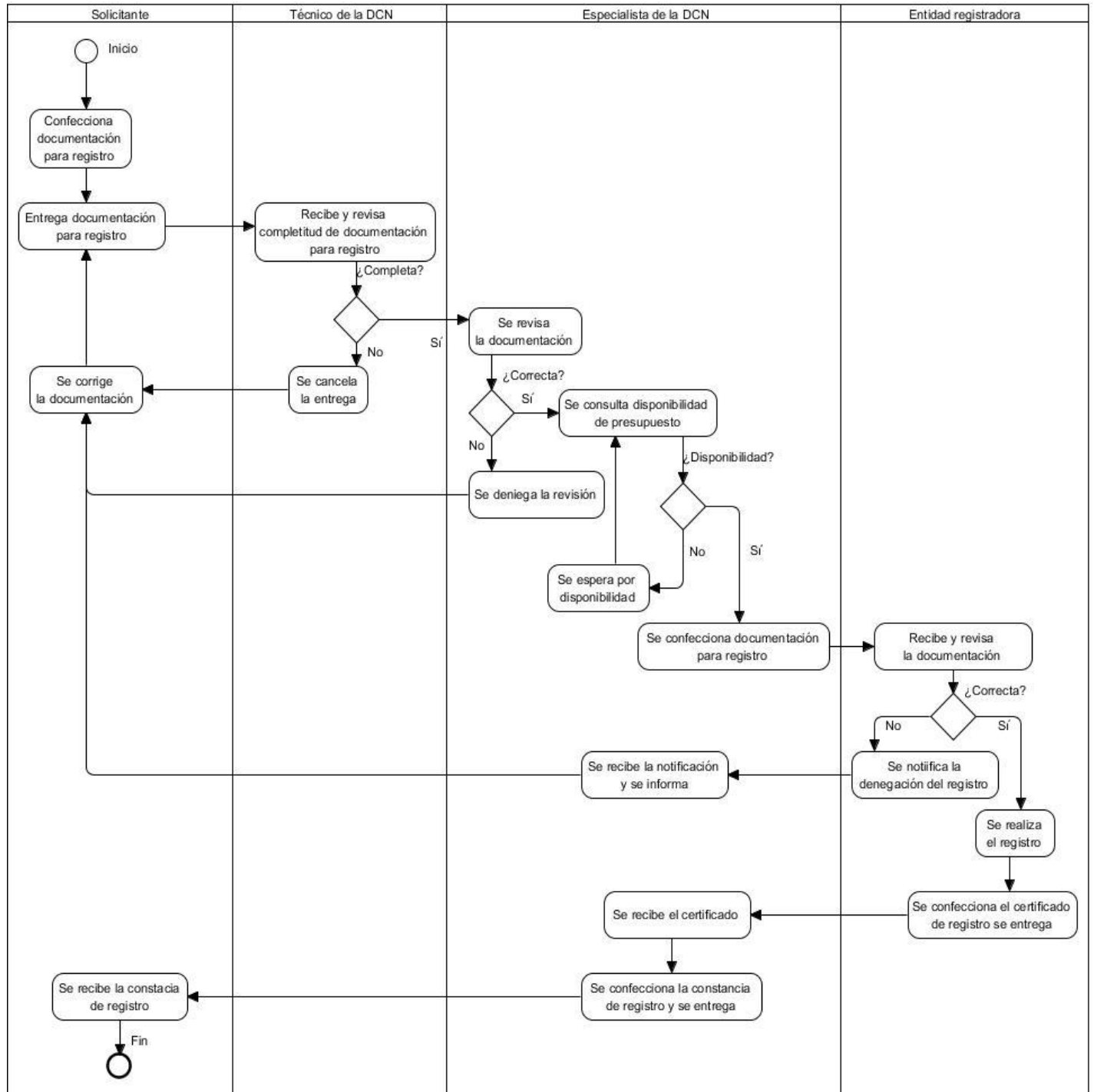
En el capítulo se realiza una descripción completa de la solución que se propone. Para esto, se hace el modelado de procesos del registro de la propiedad intelectual en la UCI. Se especifican los requisitos funcionales a cumplir por la solución y los no funcionales que presenta el sistema que se desarrollará.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

### **2.1 Proceso 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 UCI. 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 incluye 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. Después, 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 de la Dirección para la Comercialización y los Negocios crea el expediente de registro, luego el especialista es 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, posteriormente, esta entidad procede a confeccionar el certificado, que a continuación es entregado el certificado de registro al encargado del DCN, quien lo archiva y genera un nuevo certificado para entregar al autor o interesado del registro.

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

**2.2.1 Modelado de procesos del negocio**



**Figura 2 Diagrama de proceso de negocio.**

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### **2.2.2 Reglas de negocio**

Las reglas de negocios definen y controlan la estructura, el funcionamiento y la estrategia de una organización; pueden estar formalmente definidas en manuales de procedimiento, contratos o acuerdos. (Microsoft, 2014). La gestión de la propiedad intelectual en la UCI presentan diferentes reglas que permiten la ejecución del proceso con mayor calidad, estas se describen a continuación:

- La documentación que se necesita para el registro debe ser revisada y validada por la DCN antes de ser enviada a las entidades registradoras correspondientes.
- Se reconoce el derecho de autor a todos los autores de la obra con independencia del nivel de participación y la entidad de procedencia.
- El registro de una obra posee un certificado nacional.
- Se otorga el certificado de registro a una obra cuando esta es registrada por la OCPI o por el CENDA.
- Solo el DCN es el responsable en la UCI de ejecutar el proceso de registro de obras.
- Para cada registro realizado por las entidades registradoras, la DCN recibe un certificado y genera una única constancia de registro al área dentro de la UCI que solicitó el registro.
- Las solicitudes de registro de una obra se realizan por una única área dentro de la UCI.
- Las solicitudes que se envían a las entidades registradoras deben cumplir el formato establecido por dichas entidades.
- Las solicitudes y salidas del proceso de registro dentro de la UCI deben cumplir con lo establecido por la DCN para el registro de las creaciones UCI.
- Cada tipo de obra tiene definida la documentación que se necesita para su registro.
- Solo se ejecuta el registro de una obra si existe disponibilidad en el presupuesto asignado a la DCN para efectuar dicho registro.
- Las entidades registradoras definen el precio que debe abonar la UCI para efectuar el registro de las diferentes creaciones.
- Cada tipo de obra en cada entidad registradora tiene un precio para el registro.

### **2.2.3 Base legal**

En el desarrollo de la solución de software tuvo un gran peso la base legal en la que se fundamenta el proceso de registro de la propiedad intelectual en la Universidad. Como guía fundamental se estudiaron las normas y procedimientos que rigen y estandarizan este proceso en la UCI. Estas normas y procedimientos incluyen lo estipulado por las entidades registradoras para efectuar el registro de las

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

obras, pero fueron también consultados de forma individual para una mejor comprensión y para el logro de un mejor diseño de la solución.

Consultadas de la Universidad:

- IPP-1000:2008 Elaboración y aprobación de los procedimientos productivos.
- IPP-4000:2008 Registro de la propiedad intelectual.
- IPL-4020:2009 Lineamiento declaraciones legales en las creaciones UCI.

Otras consultadas:

- 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.
- Ley 14 de 28 de diciembre de 1977, del Derecho de Autor, de la Asamblea Nacional.
- Resolución 13 de 20 de febrero de 2003, del Ministro de Cultura, establece el Registro Facultativo de Obras Protegidas y de Actos y Contratos referidos al Derecho de Autor, con su respectiva reglamentación (CERLALC, 2003).
- Resolución No. 33/2008, sobre el Sistema de Registro de producto de Software, del Ministerio de la Informática y las Comunicaciones.

### **2.2 Propuesta del sistema**

El presente trabajo propone la implementación de un módulo desarrollado con tecnologías web que posibilite registrar la información que se genera como parte del proceso, registrando y posibilitando el control de la misma. Se propone la implementación del módulo Propiedad Intelectual de la Universidad de las Ciencias Informáticas. El módulo está diseñado para integrarse al SGU en el subsistema Producción que pertenece a dicho sistema.

El SGU cuenta con información centralizada y segura, que favorece considerablemente a la veracidad de las estadísticas que se obtienen. La calidad de las estadísticas apoya en gran medida el proceso de análisis de la información obtenida del sistema, ya que aporta una herramienta para la toma de decisiones en función de mejorar la calidad de los procesos de formación, extensión e investigación de la



## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

Universidad. Aumentar la calidad de los procesos sustantivos universitarios repercute directamente en la calidad de los egresados y el fortalecimiento de la formación de los profesionales de la institución. El sistema, al ser una aplicación web, facilita el trabajo de los usuarios del mismo, ya que pueden hacer uso del sistema desde cualquier área geográfica de la Universidad, viéndose reflejada la flexibilidad de la solución.

El módulo que se desarrolla, cuenta con seis agrupaciones funcionales: contabilidad, solicitud, solicitud a entidad, registro, configuración y reportes. Las agrupaciones se realizaron teniendo en cuenta el objetivo de las funcionalidades que engloban.

La agrupación funcional solicitud incorporan en el sistema las diferentes solicitudes para el registro de las obras, permitiendo ver el estado de las solicitudes a lo largo de todo el proceso. Incluye los eventos de revisión y aprobación de la documentación que se necesita para efectuar el registro en las diferentes entidades registradoras, antes de realizar las solicitudes a dichas entidades. Al proceso de solicitud podrán adjuntarse los documentos que se definan como complemento de registro, incluirse los autores de las obras, los organismos cotitulares de las obras y extraerse los datos de SGU. Si alguna de las obras fue denegada por problemas con los datos de la solicitud o con la documentación, el solicitante podrá corregirla teniendo en cuenta las sugerencias hechas por el especialista de DCN.

En solicitud a entidad se pueden realizar las diferentes solicitudes de registro a sus respectivas entidades registradoras, generándose desde el sistema las solicitudes en el formato establecido por las entidades, quedando listas para su impresión y posterior firma. Se brinda la información relacionada con el presupuesto disponible para la ejecución de los registros, permitiendo generar solamente aquellas solicitudes que puedan ser ejecutadas según la disponibilidad.

Contabilidad: incluye la gestión del presupuesto designado a la UCI por año para la ejecución del registro de las obras en las diferentes entidades registradoras. Se realiza la gestión de las diferentes monedas en las que puede estar asignando. Posibilita la asignación del precio del proceso de registro para las diferentes obras. Además, permite conocer la ejecución del presupuesto del año que transcurre.

Registro: permite la incorporación de los datos de los registros una vez registrados en las entidades. Se recogen básicamente los datos que relacionan en los documentos emitidos como parte del registro y permite la generación de constancia de registro al solicitante inicial.

Configuración: permite gestionar los datos que se necesitan para la ejecución de las diferentes funcionalidades dentro del proceso, algunos son: entidades registradoras, entidades cotitulares de obras, tipos de obras, tipologías de obras, documentos que se necesitan para el registro en dependencia de los tipos de obras, plantillas para documentos, etc.

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

Además, se desarrollan reportes que relacionan la selección de parámetros para la confección de los mismos, algunos de ellos son: consumo de presupuesto, cantidad de obras registradas por tipo y año, obras con períodos de vigencia próximo a vencerse, relación de obras por categorías que se han registrado de una determinada área de la UCI y otros que puedan confeccionarse a través de la selección de parámetros.

The screenshot displays the 'Propiedad Intelectual' module interface. At the top, a navigation bar includes icons for home, search, and help, alongside the text 'SISTEMA DE GESTIÓN UNIVERSITARIA'. Below this, a secondary navigation bar lists various university services: 'Pregrado · Postgrado · Cooperación · Residencia · Teleformación · Biblioteca · Producción · Tecnologías · Investigaciones · Extensión · Ingreso · Egreso'. The 'Propiedad Intelectual' module is highlighted in a dropdown menu. The main content area is titled 'Módulo "Propiedad Intelectual"'. On the right side, a vertical sidebar contains several menu items, each with a sub-list of options:

- Solicitud**
  - Solicitudes
  - Revisión
  - Aprobación
  - Autor externo
- Solicitud a entidad**
  - CENDA
  - OCPI
- Contabilidad**
  - Presupuesto
  - Tipo moneda
  - Monto por actividad
- Reportes**
  - Registros
- Reportes**
  - Registros de obras
- Configuración**
  - Tipo de obra
  - Tipología
  - Entidad de registro
  - Entidad titular
  - Tipo documento
  - Rol productivo
  - Configuración de plantilla

The footer of the page features the UCI logo (Universidad de las Ciencias Informáticas), copyright information for CENIA (© 2013 CENIA | Términos de uso Universidad de las Ciencias Informáticas), and the text 'Sistema de Gestión Universitaria'.

Figura 3 Módulo Propiedad Intelectual.

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### **2.3 Integración de la propuesta de solución**

El sistema de gestión de la propiedad intelectual se integra al Sistema de Gestión Universitaria (SGU), solución informática que sirve de soporte a los procesos sustantivos que se desarrollan en la UCI, algunos de ellos son: pregrado, postgrado, investigación, residencia y cooperación. Esta decisión se fundamentó en la necesidad de incorporar el trabajo con la propiedad intelectual, como uno de los procesos claves de la Universidad, en la sección de Producción. La integración de ambos sistemas propicia el intercambio de información.

El SGU cuenta con un núcleo que centraliza en componentes las funcionalidades que necesitan los procesos sustantivos integrados en ella. Estos componentes o funcionalidades horizontales permiten a las soluciones, hacer uso de ellos de una manera sencilla, lográndose una integración en poco tiempo, disminuyendo el período de desarrollo e incorporando funcionalidades estables y configurables. Los componentes o funcionalidades horizontales son:

- Configuración: gestiona la información necesaria para realizar las configuraciones del sistema, tanto globales, como las que son usadas por más de una línea de procesos. Además, permite realizar acciones de instalación o desinstalación de líneas de procesos o módulos.
- Estructura y composición: gestiona la información referente a las estructuras administrativas y su jerarquía dentro de la institución.
- Seguridad: gestiona la seguridad, los usuarios y permisos al sistema.
- Personal: gestiona la información nominal de las personas que involucran los diferentes procesos sustantivos, garantizando para todos ellos una información siempre actualizada.

### **2.4 Ingeniería de requisitos**

La ingeniería de requisitos es el proceso de descubrir, analizar, documentar y verificar los servicios y restricciones. Los requisitos, reflejan las necesidades de los clientes de un sistema que ayudan a resolver un problema determinado. De igual forma 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).

#### **2.4.1 Técnicas de obtención de requisitos**

La obtención de requisitos es el proceso mediante el cual los clientes 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 es necesario

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

por parte de los analistas el empleo de técnicas que permitan establecer una buena comunicación con los interesados del producto y así, lograr su satisfacción. A continuación se enuncian las principales técnicas utilizadas durante el proceso de desarrollo para recopilar los requisitos de software:

**Entrevistas:** se realizaron conversatorios con las personas involucradas en el negocio, principalmente los técnicos y especialistas del grupo de la Dirección para la Comercialización y los Negocios de la Universidad de las Ciencias Informáticas, con el objetivo de entender el dominio del problema y sus necesidades.

**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 (Sommerville, 2005).

### **2.4.2 Requisitos funcionales**

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este 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 funcionalidades que debe tener el módulo Propiedad Intelectual:

Requisitos Funcionales (RF)	Prioridad
RF1_Crear solicitud de registro	Alta
RF2_Modificar solicitud de registro	Alta
RF3_Mostrar lista de solicitudes	Alta
RF4_Ver detalles de solicitud de registro	Baja
RF5_Crear autor externo	Alta
RF6_Modificar autor externo	Alta
RF7_Mostrar lista autor externo	Alta
RF8_Crear presupuestos por año	Media
RF9_Modificar presupuestos por año	Media
RF10_Ver detalles de presupuestos por año	Baja
RF11_Listar presupuestos por año	Baja
RF12_Crear tipo de moneda	Media

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

RF13_Modificar tipo de moneda	Media
RF14_Ver detalles de tipo de moneda	Baja
RF15_Mostrar listado de tipo de moneda	Media
RF16_Crear monto por proceso en entidad	Media
RF17_Modificar monto por proceso en entidad	Media
RF18_Ver detalles de monto por proceso en entidad	Media
RF19_Mostrar monto por proceso en entidad	Baja
RF21_Modificar roles productivos	Media
RF22_Ver detalles de roles productivos	Baja
RF23_Mostrar roles productivos	Baja
RF24_Crear tipo de obra	Alta
RF25_Modificar tipo de obra	Media
RF26_Ver detalles de tipo de obra	Baja
RF27_Mostrar tipo de obra	Media
RF28_Crear tipología	Baja
RF29_Modificar tipología	Baja
RF30_Ver detalles de tipología	Baja
RF31_Mostrar tipología	Baja
RF32_Crear documentos de solicitudes	Media
RF33_Modificar documentos de solicitudes	Baja
RF34_Ver detalles de documentos de solicitudes	Baja
RF35_Listar documentos de solicitudes	Baja
RF36_Crear entidad cotitular	Alta
RF37_Modificar entidad cotitular	Media
RF38_Ver detalles de entidad cotitular	Media
RF39_Mostrar entidad cotitular	Media
RF40_Crear entidades registradoras	Alta
RF41_Modificar entidades registradoras	Media
RF42_Ver detalles de entidades registradoras	Baja
RF43_Listar entidades registradoras	Media

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

RF44_Incluir documentos adjuntos a la solicitud	Media
RF45_Asociar autores a solicitud	Media
RF46_Asociar documentos a solicitud	Alta
RF47_Asociar entidad titular a solicitud	Media
RF48_Mostrar esquema de búsqueda para autor externo	Alta
RF49_Revisar solicitudes con estado solicitado	Media
RF50_Incluir comentarios a solicitudes de registro con estado solicitado	Media
RF51_Listar solicitudes de registro con estado solicitado	Media
RF52_Incluir documentación dura a la solicitud	Alta
RF53_Seleccionar actividad que se realizarán con las solicitudes	Media
RF54_Listar solicitudes para CENDA con estado revisada	Media
RF55_Listar solicitudes para OCPI con estado revisada	Media
RF56_Exportar solicitud de registro en el CENDA	Alta
RF57_Exportar solicitud de registro en el OCPI	Alta
RF58_Listar solicitudes para OCPI con estado listo para registro	Media
RF59_Listar solicitudes para CENDA con estado listo para registro	Media
RF60_Consultar estado de presupuesto de las solicitudes con estado listo para registro	Alta
RF61_Listar solicitudes con estado registrado	Media
RF62_Adjuntar certificado del CENDA a solicitud con estado registrado	Media
RF63_Adjuntar certificado del OCPI a solicitud con estado registrado	Media
RF64_Crear plantilla CENDA asociado a los datos de la solicitud	Media
RF65_Crear plantilla OCPI asociado a los datos de la solicitud	Media
RF66_Exportar certificado CENDA para el solicitante	Alta

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

RF67_Exportar certificado OCPI para el solicitante	Alta
RF68_Listar configuraciones de plantillas	Media
RF69_Crear configuración de plantilla CENDA	Media
RF70_Crear configuración de plantilla OCPI	Media
RF71_Crear configuración de plantilla Constancia de registro	Media
RF72_Generar reporte consumo de presupuesto por año	Alta
RF73_Generar reporte cantidad de registro dado una fecha	Alta

**Tabla 1 Requisitos funcionales.**

### 2.4.3 Requisitos no funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Además, 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 Propiedad Intelectual:

<b>Usabilidad</b>	
RNF_1	El sistema debe presentar un menú lateral y una barra de iconos flotantes que permitan el acceso rápido a la información por parte de los usuarios, aprovechando así las potencialidades de estas estructuras.
<b>Apariencia e interfaz externa</b>	
RNF_2	Las vistas del sistema deben indicar en cada momento la acción que se está realizando así como lo iconos deben estar representados por una imagen acorde a la acción que se realiza.
RNF_3	Las interfaces deben cumplir con las pautas de diseño definidas para el SGU.
<b>Seguridad</b>	
RNF_4	Solo se mostrarán a los usuarios aquellas acciones o informaciones a las que por su responsabilidad o rol dentro del negocio necesitan acceder, mostrando en la vista mediante iconos y vínculos el acceso a la misma.
RNF_5	El servidor de aplicaciones y de base de datos deberá mantener una seguridad mediante firewall para proteger el código y la información.
<b>Software</b>	

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

RNF_6	Para el despliegue del sistema se contará en el servidor de base de datos con Postgres 9.1 o superior bajo el sistema operativo Ubuntu Server 12.04 o superior.
RNF_7	Para el despliegue del sistema estará instalado en el servidor de aplicaciones web con el sistema operativo CentOS 6.2 donde se encuentre instalado PHP v5.3 con las librerías php5-ldap, php5-gd, php5-mcrypt, php5-pgsql, php5-xsl, php5-openssl, Apache 2.2 con el módulo rewrite activado y JDK 6 o superior.
RNF_8	Para el uso del sistema se requiere una PC cliente con el navegador web Mozilla Firefox 17 o superior.
<b>Hardware</b>	
RNF_9	Para la ejecución del sistema se requiere que la PC cliente tenga los siguientes componentes de hardware: Pentium 4 o superior, 512 MB RAM y 1 GB disco duro disponible como mínimo.
RNF_10	El servidor de aplicación y el de base de datos deberán tener los siguientes componentes de hardware para su funcionamiento en el ambiente en que se encuentra desplegado el SGU: Microprocesador de 8 núcleos, 4GB de memoria RAM, 250 GB disco duro.
<b>Confiabilidad</b>	
RNF_11	Cuando ocurra una excepción, el sistema mostrará un mensaje explicativo del error ocurrido y permanecerá en el mismo estado sin realizar ninguna otra operación.
<b>Eficiencia</b>	
RNF_12	La solución propuesta debe soportar una conexión simultánea de 100 usuarios.
RNF_13	El sistema deberá tener por cada transacción un tiempo de respuesta promedio de 1,5 segundos y un máximo de 5 segundos.
<b>Soporte</b>	
RNF_14	El sistema contará con toda la documentación definida en el expediente de proyecto asociada a su proceso de desarrollo para las actividades de soporte.
RNF_15	Todos los productos de trabajo generados en el desarrollo del software se regirán por las pautas de configuración debidamente documentadas en el expediente de proyecto.
<b>Legales</b>	
RNF_16	IPP-1000:2008 Elaboración y aprobación de los procedimientos y lineamientos para la actividad productiva.
RNF_17	IPP-4000:2008 Registro de la propiedad intelectual.
<b>Restricciones de diseño e implementación</b>	
RNF_118	El software deberá ser desarrollado en su totalidad con tecnologías y componentes de



*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

código abierto.
-----------------


**Tabla 2 Requisitos no funcionales.**

**2.4.4 Descripción de requisitos**

Es el producto del trabajo final que genera la Ingeniería de Requisitos. Describe la función y el desempeño de un sistema basado en computadoras y las restricciones que regirán su desarrollo (Pressman, 2002).

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFPI21	Modificar roles productivos	<p>El requisito permite modificar los roles productivos que tendrán los autores de las obras.</p> <p>El requisito comienza cuando el administrador del sistema selecciona del módulo <b>Propiedad Intelectual</b> del subsistema <b>Producción</b>, la opción <b>Rol productivo</b> del menú de funcionalidades <b>Configuración</b>.</p> <p>El administrador del sistema selecciona la opción <b>Modificar</b> en el listado de roles productivos mostrándose los datos correspondiente a los roles productivos (todos los datos pueden modificarse):</p> <ul style="list-style-type: none"> <li>- Nombre del rol (obligatorio)</li> <li>- Descripción</li> <li>- Habilitado</li> </ul> <p>Una vez especificado los datos el administrador selecciona la opción <b>Aceptar</b>, mostrándose el mensaje: El elemento ha sido modificado satisfactoriamente. Guardándose los datos del rol productivo modificado y luego se muestra la lista de roles. Se muestra en el área de iconos flotantes el icono <b>Listar</b>.</p>	Baja	Media
<b>Prototipo</b>				

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

	<div style="border: 1px solid #ccc; padding: 10px;"> <p style="text-align: right;"></p> <p><b>Modificar rol</b></p> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 45%;"> <p>Nombre rol: <span style="color: red;">*</span></p> <input type="text" value="Analista"/> </div> <div style="width: 45%;"> <p>Descripción:</p> <div style="border: 1px solid #ccc; height: 80px; width: 100%;"></div> </div> </div> <p><input checked="" type="checkbox"/> Habilitado</p> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </div> </div>		
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>
	Nombre del rol	<ul style="list-style-type: none"> <li>varchar</li> </ul>	<ul style="list-style-type: none"> <li>Obligatorio</li> <li>Admite entre 2 y 50 caracteres, solo espacios y guión bajo entre palabras</li> </ul>
	Descripción	<ul style="list-style-type: none"> <li>varchar</li> </ul>	<ul style="list-style-type: none"> <li>Admite entre 5 y 1000 caracteres. La cantidad de caracteres permitidos para una palabra es de 30.</li> </ul>
	Habilitado	<ul style="list-style-type: none"> <li>boolean</li> </ul>	<ul style="list-style-type: none"> <li>Campo de selección que permite especificar si un rol productivo está habilitada o no.</li> </ul>
	<b>Observaciones</b>	<ol style="list-style-type: none"> <li>Para realizar la acción el administrador tiene que estar autenticado en el sistema.</li> <li>En caso que el elemento ya exista se muestra un mensaje de error: El elemento ya existe. El elemento existe si coincide el nombre del rol.</li> <li>En caso del administrador seleccionar la opción Cancelar se muestra el mensaje: ¿Está seguro que desea realizar la acción? Luego si selecciona Aceptar se cancela la acción y se vuelve al listado de roles, si selecciona Cancelar desaparece el mensaje y se mantiene la interfaz de crear.</li> <li>En caso del nombre del rol tener menos de 2 caracteres se muestra el mensaje en rojo: Entre al menos 2 caracteres.</li> </ol>	

## Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

		<ol style="list-style-type: none"><li>5. En caso de introducir caracteres extraños en el nombre, se muestra un mensaje en rojo: Entre solo letras, números y espacio o guión bajo entre palabras.</li><li>6. En caso que se deje un campo obligatorio vacío se muestra un mensaje de error: Campo requerido y se muestra en rojo el campo que debe ser llenado obligatorio.</li><li>7. En caso de la descripción tener menos de 5 caracteres, se muestra el mensaje: Entre al menos 5 caracteres.</li></ol>
--	--	---

Tabla 3 Descripción del requisito modificar roles productivos.

### 2.5 Arquitectura del sistema

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente: al que inicia el diálogo o solicita los recursos y servidor: al que responde a las solicitudes. Es el modelo de interacción más común entre aplicaciones en una red (Instituto Tecnológico de Colima, 2010). Una representación gráfica de este tipo de arquitectura se muestra a continuación:

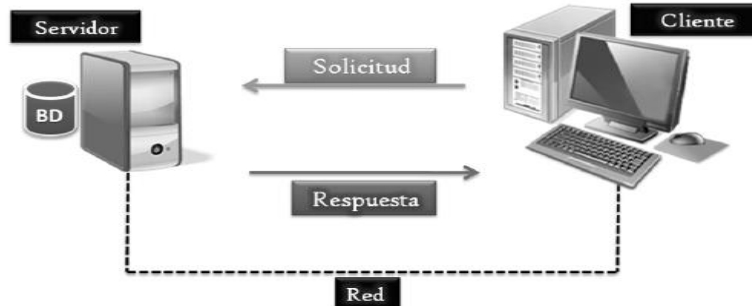


Figura 4 Arquitectura cliente servidor.

### 2.6 Patrones

#### 2.6.1 Patrón de la arquitectura

El Modelo-Vista-Controlador (MVC) es un patrón de desarrollo o un estilo de arquitectura de software que separa el código fuente de las aplicaciones en tres grupos:

**Modelo:** todo el código que tiene que ver con el acceso a base de datos. En el modelo se mantiene encapsulada la complejidad de la base de datos y simplemente 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 pueden invocar las funciones que se necesiten del modelo y este se encargará de procesarlas. En el desarrollo del sistema las clases modelos y las librerías

## Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

se encuentran en la capa Modelo. Las librerías son las encargadas de organizar la lógica del negocio, mientras que los modelos son los que acceden a los datos. Las clases librerías envían los datos a las controladoras después de obtenerlos de los modelos.

**Vista:** la vista codifica y mantiene la presentación final de cara al usuario, es decir, en la vista se coloca todo el código HTML, CSS, *Javascript*, etc; que se tiene que generar para producir la página que se quiera que vea el usuario.

**Controlador:** la controladora funciona como enlace entre el modelo, la vista y cualquier otro recurso que se tenga que procesar en el servidor para generar la página web (Álvarez, 1999). En el sistema de propiedad intelectual es el intermediario entre la vista y el modelo. Es la clase encargada de recibir las peticiones del usuario y en vista de satisfacer su demanda, utiliza 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. La Figura 5 muestra el funcionamiento del patrón MVC en el marco de trabajo GUUD.

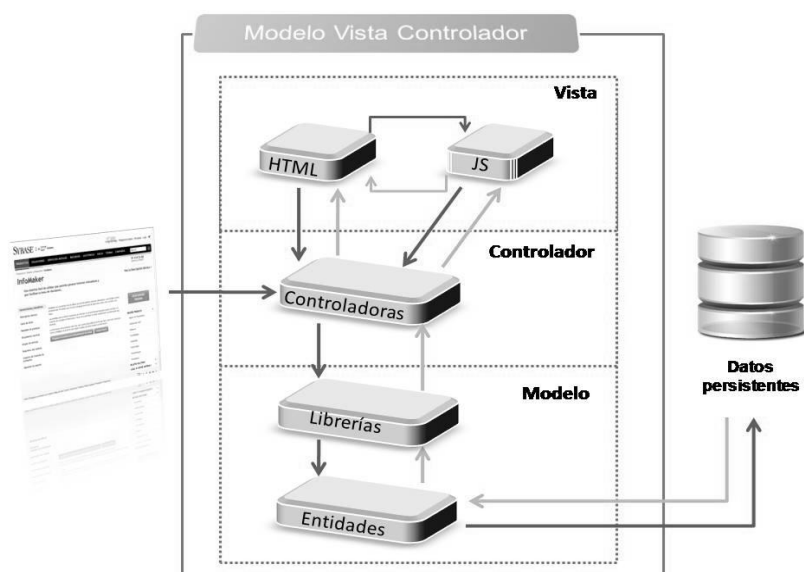


Figura 5 Funcionamiento del patrón Modelo-Vista-Controlador en G.U.U.D..

### 2.6.2 Patrones del diseño

#### Patrones GRASP

Los Patrones Generales de Software para Asignación de Responsabilidades (GRASP por sus siglas en inglés) describen los principios fundamentales del diseño de objetos para la asignación de

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable (Astudillo, Marcello Visconti y Hernán, 2003). Para el diseño del módulo Propiedad Intelectual, 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.

**Experto:** asigna una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. En el módulo Propiedad Intelectual se evidencia en las clases con categoría de librería. Por ejemplo: la clase `solicitud_lib`, es la que cuenta con la información necesaria para cumplir las responsabilidades sobre los elementos de negocio.

**Creador:** asigna a la clase B la responsabilidad de crear una instancia de clase A. Por ejemplo: en la clase `loader`, objeto `load` dentro de las clases controladoras, se encarga de acceder a los elementos del marco de trabajo como son: librerías, modelos y vistas; en el módulo seguridad, la librería `fabrica_ma_lib`, que se encarga de crear los objetos de los modos de autenticación.

**Controlador:** asigna la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Las clases con la categoría de controlador son las encargadas de obtener datos y enviar estos hacia las librerías y las vistas.

**Bajo acoplamiento:** asigna responsabilidades a las clases para mantener bajo acoplamiento entre ellas. El grado de acoplamiento no puede considerarse aisladamente de otros principios como Experto y Alta Cohesión, sin embargo, es un factor a considerar cuando se intente mejorar el diseño. Una evidencia del patrón en el trabajo lo constituye, la definición de la clase controladora asociada a la solicitud de registro, la cual tiene la responsabilidad de gestionar únicamente la información de la solicitud de registro.

**Alta cohesión:** asigna responsabilidades a las clases de modo que la cohesión entre ellas siga siendo alta. Evidenciándose en el funcionamiento de la clase controladora `solicitud` al solicitar a la clase controladora `documento`, los documentos asociados a ella. La clase controladora `solicitud` no conoce sus documentos asociados, pero al relacionándose con la clase controladora `documento` puede obtener la que necesita.

### **Patrones GoF**

Para el diseño del módulo Propiedad Intelectual se aplican los patrones la Banda de los Cuatros (GoF por sus siglas en inglés), los creacionales: instancia única y los de comportamiento: mediador y observador. Los patrones creacionales crean instancias de objetos, el objetivo de estos es abstraer el proceso de instanciación y ocultar los detalles de cómo el objeto es creado o inicializado, mientras que los patrones

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

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.

**Instancia única:** garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Se evidencia en todas las clases controladoras que son instancias únicas y la IoC para la interacción entre el módulo Propiedad Intelectual y los otros módulos a los que se integra.

**Mediador:** define un objeto que coordine la comunicación entre objetos de distintas clases del Módulo, pero que funcionan como un conjunto. Se refleja en las librerías, que funcionan como mediadoras entre las clases controladoras y las modelos o acceso a datos.

**Observador:** define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él (Astudillo, Marcello Visconti y Hernán, 2003). Este patrón se evidencia en la clase *loader* que es el objeto load de las clases controladoras existentes en el módulo Propiedad Intelectual, esta clase, se encarga de cargar los elementos del marco de trabajo dígame librerías y modelos que se encargan de actualizar la controladora instanciada.

### **2.6.3 Patrones de base de datos**

#### **Patrón de llaves subrogadas**

El patrón llave subrogada facilita la interacción con la Base de Datos en un futuro. El mismo plantea que se debe generar una llave primaria única para cada entidad, en vez de usar un atributo identificador en el contexto dado. Esto permite que las tablas sean más fáciles de consultar a partir del identificador, pues todos tienen el mismo tipo en cada una de las tablas. Por ejemplo: la entidad Solicitud, que representa la solicitud para el registro de una obra, un identificador de esta entidad debería estar constituido por el nombre del producto, nombre del proyecto, versión y fecha de registro; sin embargo, aplicando este patrón se adicionaría un atributo id de tipo autoincremental, que sería la llave primaria para esta entidad, en vez de la combinación de todos aquellos atributos.





## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### **2.8 Diagrama de despliegue**

Un diagrama de despliegue muestra cómo se configuran las instancias de los componentes y los procesos para un tiempo de ejecución en las instancias de los nodos de proceso. (Larman, 2001)

El módulo 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 obtener los datos. La interacción al sistema se realiza mediante un navegador web de una PC cliente, se utiliza para esto el protocolo HTTPS.

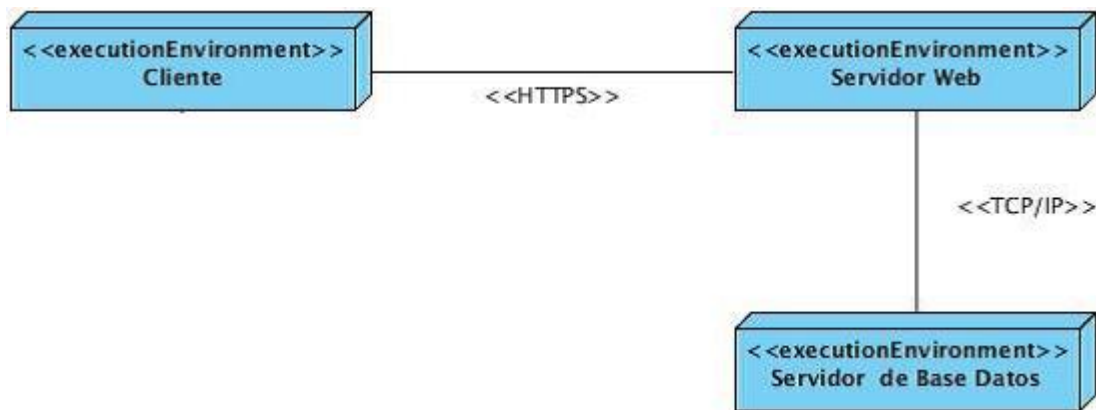


Figura 7 Diagrama de despliegue.

#### **Descripción de los elementos e interfaces del modelo de despliegue**

- **Computadora Cliente:** su función es acceder al sistema e interactuar con el mismo según sus necesidades. Al estar la aplicación desarrollada sobre la web, la máquina cliente necesita disponer de muy pocas prestaciones, solo se necesita un navegador web para poder acceder al sistema y realizar las operaciones necesarias.
- **Servidor de Aplicaciones:** ordenador donde se encuentra el servidor web Apache, este será el lugar en que se gestione todo el contenido de la aplicación y las máquinas clientes acceden a través de un navegador web.
- **Servidor de Base de Datos:** es el encargado de almacenar toda la información generada del sistema y dar respuesta a las peticiones hechas por la aplicación.
- **Protocolo HTTPS:** protocolo utilizado para la conexión entre las computadoras cliente y el servidor web.
- **Protocolo TCP/IP:** familia de protocolos utilizado para la conexión entre el servidor web y el servidor donde se encuentra ubicada la base de datos.



## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### **Conclusiones parciales**

En el presente capítulo se realizó un análisis profundo del negocio, donde se describen las actividades para el registro de la propiedad intelectual, y esto permitió la obtención del conjunto de funcionalidades que debe cumplir el sistema quedando especificados los requisitos del software. A partir de los artefactos generados, se obtuvo el análisis y diseño del módulo Propiedad Intelectual del Sistema de Gestión Universitaria, que sirven como entrada a la etapa de implementación y prueba de la solución.

# *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

## **Capítulo 3: Implementación y prueba de la solución propuesta**

### **Introducción**

En la fase de desarrollo de sistemas, la implementación y validación de la solución trae como resultado el mejoramiento del proceso de la propiedad intelectual. En el presente capítulo, se describe la implementación de la solución propuesta descrita en el capítulo anterior, teniendo en cuenta un estándar de codificación para la implementación de las funcionalidades, posibilitando la legibilidad del código. Además, se presentan las pruebas para verificar el correcto funcionamiento del sistema y descubrir los errores cometidos al realizar la construcción del sistema.

### **3.1 Estándar de codificación**

El código debe ser desarrollado siguiendo estándares de desarrollo para facilitar su lectura y la modificación por cualquier miembro del equipo de desarrollo. Para poder llevar a cabo con éxito la propiedad colectiva del código esta práctica es decisiva. La propiedad colectiva sería impensable sin una codificación basada en estándares que haga que todo el mundo se sienta cómodo con el código escrito por cualquier otro miembro del equipo.

Para el desarrollo del módulo Propiedad Intelectual se utiliza el estándar de codificación establecido por la Dirección de Informatización siguiendo las pautas del diseño establecidas en el manual de directrices del Sistema de Gestión Universitaria.

#### **3.1.1 Identación, llaves de apertura y cierre, y tamaño de las líneas**

Usar una indentación sin tabulaciones, con un equivalente a 4 espacios. 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 obtenerEntidadRegistroDadoIdPadre()
{
    $id = $this->input->all_post();
    $entidadRegistro =$this->solicitud_lib->obtenerEntidadRegistroDadoIdPadre($id['id']);
    echo json_encode($entidadRegistro);
}
```

**Figura 8** Indentación y llaves.

## Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

### 3.1.2 Convención de nomenclatura

**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.

```
$estado :
$entidadRegistro
```

Figura 9 Nomenclatura para crear variables.

**Clases:** siempre comienzan con mayúscula, en caso de nombre compuesto las palabras se separan con el carácter subrayado “\_” y el resto en minúscula.

```
class Presupuesto_tipo_obra extends MY_Controller
```

Figura 10 Nomenclatura para crear clases.

**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.

```
public function modificarSolicitudExterna()
{
    $all_post = $this->input->all_post(TRUE);
    $response = $this->registro_lib->modificarSolicitud($all_post);
    if ($response != false)
    {
        $this->message('SYS001');
    }
    else
    {
        throw new Exception_Error('SYS005');
    }
}
```

Figura 11 Nomenclatura para crear funciones.

**Ficheros:** todo siempre en minúscula y en caso de nombres compuestos se usa el carácter subrayado “\_”.

1. **Vistas:** intuitivo y relacionado con el formulario y/o vista que representa.
2. **Modelos:** con el mismo nombre de la clase que representa que contiene en el nombre el sufijo

## Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

\_mdl o \_base en caso de ser modelos base.

3. **Librerías:** con el mismo nombre de la clase que representa que contiene en el nombre el sufijo \_lib.
4. **Controladoras:** con el mismo nombre de la clase que representa.

### 3.1.3 Estructuras de control

Se incluye entre las estructura de control (if, for, foreach, while, switch) y en los paréntesis debe existir un espacio. 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.

```
public function modificarSolicitudExterna()
{
    $all_post = $this->input->all_post(TRUE);
    $response = $this->registro_lib->modificarSolicitud($all_post);
    if ($response != false)
    {
        $this->message('SYS001');
    }
    else
    {
        throw new Exception_Error('SYS005');
    }
}
```

Figura 12 Estructura de control.

### 3.1.4 Documentación

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

```
/**
 * Clase controladora propiedad_intelectual
 *
 * Esta clase funciona como intermediaria entre la vista
 * y las librerías
 *
 * @package          Producción
 * @subpackage       Controllers
 * @category         Controllers
 * @author           Carlos Yusel Carralero Calzadilla
 * @author           Yusel Carlos Carralero Calzadilla
 */
```

Figura 13 Documentación de la clase controladora registrar.

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### **3.2 Validación de los requisitos**

Es muy importante asegurar la validez de los requisitos previamente a comenzar un desarrollo de software. Para ello debe hacerse una comprobación de la correspondencia entre las descripciones iniciales y si el modelo es capaz de responder al planteamiento inicial. Para llevar a cabo esto, se suele realizar comprobando que el modelo obtenido responde de la misma forma deseada que la que el cliente pide por un lado, y por otro a la inversa si otras respuestas del modelo convencen al cliente.

#### **3.2.1 Criterios de validación de requisitos del cliente**

El proceso de desarrollo de software utilizando la metodología DAC propone criterios que permiten validar los requisitos del cliente, donde se responde a varias interrogantes que validan si el requisito se aprueba o no.

##### **Interrogantes para validar los requisitos del cliente**

- ¿El proveedor del requisito es un proveedor válido?
- ¿El requisito está identificado como único?
- ¿El requisito es modificable?
- ¿El requisito no es ambiguo?
- ¿El requisito está completo?
- ¿El requisito es congruente con otros REQ relacionados?
- ¿El requisito puede ser implementado?
- ¿El requisito puede ser probado?
- ¿El resultado de la evaluación de impacto es positivo?
- ¿El requisito está correcto?
- ¿El requisito es traceable?

##### **Resultado de aplicar los criterios de validación**

Con la aplicación de los criterios para validar requisitos del cliente, se logró obtener el 100 % de los requisitos aprobados como se refleja en el documento: Criterios para validar requisitos del cliente en el expediente del proyecto Producción.

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### **3.2.2 Técnicas de validación de los requisitos**

- **Revisión de requisitos:** se realizaron reuniones entre varios miembros del proyecto, fundamentalmente analistas, donde se revisaron los documentos de especificación de requisitos y se detectaron un número de inconsistencias, que pueden ser anomalías, omisiones u otros tipos de errores, a las que se les dio solución.
- **Construcción de prototipos:** se mostró un modelo ejecutable del sistema a los usuarios finales y clientes para que los mismos interactuaran con este y determinaran si cumplía o no con sus necesidades.
- **Generación de casos de prueba:** se realizaron los diseños de casos de pruebas para cada uno de los requisitos especificados, permitiendo verificar que todos se pudieran probar e identificar en medida de la complejidad del diseño de caso de prueba, requisitos que deberían ser reconsiderados y cuáles pueden ser los más difíciles de implementar (Sommerville, 2005).

### **Resultado de la revisión de los requisitos**

Durante la revisión de requisitos se realizaron verificaciones en el documento Especificaciones de requisitos. Este proceso comprendió:

- Verificaciones de validez: los requisitos deben cumplir con las necesidades del cliente. Luego de realizar algunos análisis y razonamientos surgieron funciones adicionales y cambios en las que ya estaban identificadas.
- Verificaciones de consistencia: los requisitos no deben contradecirse en las especificaciones escritas, no debe haber restricciones o descripciones que estén opuestas a las reglas definidas.
- Verificaciones de completitud: los requisitos deben incluir todas las funcionalidades propuestas por el cliente, satisfacer de manera general todas las necesidades acordadas.
- Verificabilidad: para evitar posibles discusiones entre los miembros del equipo del proyecto y el cliente, se revisó que los requisitos estuvieran descritos de manera que puedan ser verificables, o sea, que se puedan diseñar casos de pruebas orientados a estos y que demuestren que el sistema a entregar responde a las necesidades del cliente.

Como resultado de este proceso se identificaron inconsistencias en las especificaciones tales como:

- Falta de concordancia entre la complejidad de la especificación y la registrada en el documento de evaluación de requisitos.
- Componentes innecesarios en los prototipos de interfaz.

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

- Descripciones poco detalladas de algunos de los requisitos.
- Errores ortográficos.

### **3.3 Pruebas de software**

En la investigación se utilizan algunas de las pruebas definidas por *Pressman* en su libro *Ingeniería de software: Un enfoque práctico*, donde el objetivo de la prueba del software es descubrir errores; se cumple planeando y ejecutando una serie de pasos (pruebas de unidad, integración, validación y sistema). Las pruebas de unidad e integración se concentran en la verificación funcional de cada componente y en la incorporación de componentes en la arquitectura del software. La prueba de validación demuestra el cumplimiento con los requisitos del software, y la prueba del sistema valida el software una vez que se ha incorporado al sistema (Pressman, 2002). En la Tabla se muestran la estrategia pruebas diseñadas a aplicar al módulo Propiedad Intelectual para comprobar que la solución cumple con los requisitos definidos.

Nivel	Tipo	Método	Técnica
Unidad	Funcional	Caja blanca	Camino básico
Aceptación	Funcional	Caja negra	Alfa
Integración	Funcional	Caja negra	Incremental
Sistema	Rendimiento y resistencia	Caja negra	Automática
	Funcional	Caja negra	Particiones de equivalencia

**Tabla 4 Estrategia de pruebas de software.**

#### **3.3.1 Pruebas unitarias**

La prueba de unidad centra el proceso de verificación en la menor unidad del diseño del software. Se prueba la interfaz del módulo para asegurar que la información fluye de forma adecuada hacia y desde la unidad del programa que está siendo probada. Además, se examinan las estructuras de datos locales para asegurar que los datos que se mantienen temporalmente, conservan su integridad mediante los pasos de ejecución del algoritmo (Pressman, 2002). La interfaz del sistema puede ser probada mediante el método de caja negra, también denominada prueba de comportamiento, esta se centra en los requisitos funcionales del software. Las estructuras de datos pueden ser probadas mediante el método de caja

## Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

blanca o estructural, que se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa.

### Pruebas de caja blanca o estructural

La prueba de caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que garanticen que se ejercita, por lo menos una vez, todos los caminos independientes de cada módulo, ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa, ejecuten todos los bucles en sus límites y con sus límites operacionales y ejerciten las estructuras internas de datos para asegurar su validez (Pressman, 2002).

La técnica utilizada en la investigación para realizar la prueba de caja blanca es camino básico: que permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución (Pressman, 2002).

Esta prueba se realiza de forma automática en el sistema, para esto *CodeIgniter* como parte estructural del marco de trabajo GUUD posee la librería `unit_test` para la ejecución de pruebas estructurales, aunque es bastante sencilla, cuenta con una sola función de evaluación y dos funciones de resultados, permite determinar con certeza si un código específico produce el tipo de dato y resultado esperado. Para ejecutar la prueba se utiliza la línea de código: `$this->unit->run (código, resultado esperado, 'nombre de prueba');`; Donde “código”: es el segmento de código que se desea probar; “resultado esperado”: es lo que debe devolver la evaluación del código, que puede ser un tipo de dato o un valor literal y “nombre de prueba”: es el nombre que se le puede dar a la prueba.

A continuación se muestra un ejemplo de las pruebas de caja blanca realizadas.

<b>Prueba estructural de caja blanca</b>	<b>Código caso de prueba:</b>
	Registrar entidad
<b>Probador:</b>	Yusel Carlos Carralero Calzadilla
<b>Tipo de dato esperado:</b>	Boolean



Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

Código al que se aplica:

```
public function registrarEntidad($entidades)
{
    $this->_ci->db->trans_start();
    foreach ($entidades as $key => $value)
    {
        $esta=$this->_ci->tb_nentidad_externa_md1->obtenerDadoId(array('id_entidad_externa'=>$value));
        if(empty($esta))
        {
            $datos= array();
            $datos['id_entidad_externa']=$value;
            $this->_ci->tb_nentidad_externa_md1->registrar($datos);
        }
    }
    $this->_ci->db->trans_complete();
    return $this->_ci->db->trans_status();
}
```

Función de evaluación:

```
public function prueba(){
    $datos=array();
    $datos[0]=34;
    echo $this->_ci->unit->run($this->registrarEntidad($datos),true,"Prueba:'Registrar entidad'");
}
```

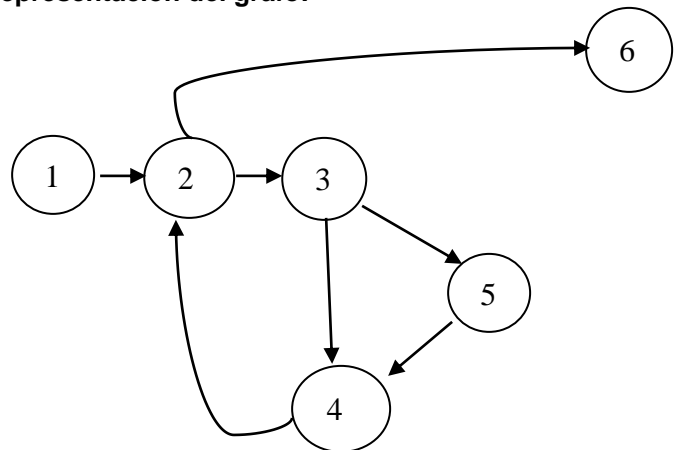
Complejidad ciclomática:

$V(G) = (A-N) + 3 = (7-6) + 2 = 3$

Caminos independientes:

- 1. 1-2-6
- 2. 1-2-3-4-2-6
- 3. 1-2-3-5-4-2-6

Representación del grafo:



Resultado de la prueba:

Test Name	Prueba:'Registrar entidad'
Test Datatype	Boolean
Expected Datatype	Boolean
Result	Passed
File Name	/var/www/tesisNucleo3/base/application/produccion/propiedad_intelectual/libraries/entidad_externa_lib.php
Line Number	48

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

<b>Evaluación del caso de prueba:</b>	Satisfactoria
---------------------------------------	---------------

**Tabla 5 Prueba de unidad registrar entidad.**

### **3.3.2 Pruebas de integración**

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. Este tipo de prueba se divide en incremental y no incremental, esta última combina todos los módulos por anticipado. Se prueba todo el programa en conjunto, donde normalmente, se llega al caos pues se encuentran un gran número de errores y la corrección se hace difícil, puesto que es complicado aislar las causas al tener delante el programa entero en toda su extensión. Una vez que se corrigen esos errores aparecen otros nuevos y el proceso continúa en lo que parece ser un ciclo sin fin. La integración incremental, el programa se construye y se prueba en pequeños segmentos donde los errores son más fáciles de aislar y de corregir, es más probable que se puedan probar completamente las interfaces y se puede aplicar un enfoque de prueba sistemática (Pressman, 2002). Por esta razón, se escogió el enfoque incremental para la realización de las pruebas de integración de la solución. Estas pruebas aparecen relacionadas en el Anexo 1. Al finalizar las pruebas de integración no fueron detectados errores asociados a la interacción del módulo Propiedad Intelectual con los módulos Seguridad, Estructura y Composición y Personal del Sistema de gestión Universitaria (SGU).

<b>Módulo al que se integra</b>	Estructura y Composición.
<b>Condiciones de ejecución</b>	En el módulo Estructura y Composición deben existir estructuras configuradas.
<b>Descripción de la prueba</b>	Comprobar que el módulo Propiedad Intelectual recibe las estructuras incorporadas en el módulo Estructura y Composición para ser seleccionadas como entidades titulares para el proceso de registro.
<b>Entradas/Pasos de ejecución</b>	Mediante la funcionalidad entidad cotitular del módulo Propiedad Intelectual se solicita al módulo Estructura y Composición las estructuras que tiene activas. Estas estructuras mostradas son seleccionadas como entidades titulares almacenándose esta información en el módulo Propiedad Intelectual.
<b>Resultado esperado</b>	Conformación de la lista de entidades titulares.
<b>Evaluación</b>	Prueba satisfactoria.

**Tabla 6 Prueba de integración con el módulo Estructura y Composición.**

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### **3.3.3 Pruebas del sistema**

La prueba del sistema, realmente, está constituida por una serie de pruebas cuyo propósito primordial es ejercitar profundamente el sistema basado en la computadora, aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas. Para este caso se realizaron las siguientes pruebas (Pressman, 2002).

#### **Pruebas de caja negra o funcional**

La prueba de caja negra se centra en los requisitos funcionales del software, permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de caja negra intenta encontrar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación. Este método se utiliza a través de la realización de los casos de prueba basados en requisitos, utilizando la técnica de particiones de equivalencia.

<b>Escenario</b>	<b>Descripción</b>	<b>Respuesta del sistema</b>	<b>Flujo central</b>
EC 1.1 Mostrar los datos correctamente	Mediante este escenario el usuario introduce todos los datos correctamente.	Muestra todos los roles creados, y la cantidad de roles que se seleccionaron en Cantidad por página, si se escribe el nombre de un rol en la opción buscar, muestra dicho rol si existe.	El requisito se ejecuta cuando el administrador del sistema selecciona del módulo Propiedad Intelectual del subsistema Producción, la opción Entidad de registro del menú de funcionalidades Configuración. Se muestra un listado con todas los roles que se han insertado, con los datos: nombre del rol con las opciones Crear en el área de íconos flotantes, Ver detalles y Modificar en el área de íconos internos.
EC 1.2 Entrar en la opción buscar un rol no creado	En este escenario el usuario introduce en la opción buscar un rol no existente	No lista ningún rol	El requisito se ejecuta cuando el administrador del sistema selecciona del módulo Propiedad Intelectual del subsistema Producción, la opción Entidad de registro del menú de funcionalidades Configuración.

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

			Se muestra un listado con todas las entidades de registro que se han insertado, con los datos: nombre de la entidad, con las opciones Crear en el área de íconos flotantes, Ver detalles y Modificar en el área de íconos internos. El usuario inserta en la opción Buscar un rol no existente.
--	--	--	---

**Tabla 7 Diseño de caso de prueba listar rol.**

Para ver los diseños de casos de pruebas en su totalidad, consultar el expediente del proyecto Producción. Al aplicar la prueba de caja negra se alcanzaron los siguientes resultados en cada una de las iteraciones, como se muestra en la Tabla 8. Las no conformidades encontradas fueron resueltas en su totalidad.

Iteraciones	No conformidades	Asociadas a
1ra	26	Errores de interfaz, de validación y ortografía.
2da	11	Errores de interfaz, de validación y ortografía.
3ra	0	-

**Tabla 8 Relación de las no conformidades.**

### **Prueba de resistencia (Stress)**

Las pruebas de resistencia están diseñadas para enfrentar a los programas con situaciones anormales. En esencia, cuando se realiza la prueba de resistencia se pregunta: ¿A qué potencia puedo ponerlo a funcionar antes de que falle? , la misma se ejecuta en un sistema de forma que demande recursos en cantidad, frecuencia o volúmenes anormales (Pressman, 2002).

### **Prueba de rendimiento (carga)**

La prueba de rendimiento está diseñada para probar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado. La prueba de rendimiento se da durante todos los pasos del proceso de la prueba. Incluso al nivel de unidad, se debe asegurar el rendimiento de los módulos individuales a medida que se llevan a cabo las pruebas de caja blanca, sin embargo, hasta que no estén completamente integrados todos los elementos del sistema, no se puede asegurar realmente el rendimiento del sistema. Las pruebas de rendimiento, a menudo, van emparejadas con las pruebas de

## Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

resistencia y, frecuentemente, requieren instrumentación tanto de software como de hardware (Pressman, 2002).

### Resultados de las pruebas de rendimiento y stress

Para la realización de estas pruebas se utilizó la herramienta JMeter 2.3.4, la misma 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 (Apache Software Foundation, 2013). A continuación se muestran en la Figura 14 los resultados de las pruebas realizadas.

Label	# Muestras	Media	Mediana	Línea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/tesisNucleo2/...	1	324	324	324	324	324	0,00%	3,1/sec	1,5
/tesisNucleo2/...	1	697	697	697	697	697	0,00%	1,4/sec	1,7
/tesisNucleo2/...	1	362	362	362	362	362	0,00%	2,8/sec	1,2
/tesisNucleo2/...	1	347	347	347	347	347	0,00%	2,9/sec	3,7
/tesisNucleo2/...	1	373	373	373	373	373	0,00%	2,7/sec	,2
TOTAL	5	420	362	697	324	697	0,00%	2,4/sec	1,6

Figura 14 Prueba de carga y stress resultado para una muestra.

### Descripción de los campos

- **Label:** el nombre de la muestra.
- **Muestras:** el número de muestras para cada URL.
- **Media:** el tiempo medio transcurrido para un conjunto de resultados.
- **Mediana:** significa que el 50% de las peticiones realizadas por n usuarios, tardaron menos del valor reflejado.
- **Mín.:** el mínimo tiempo transcurrido para las muestras de la URL dada.
- **Máx.:** el máximo tiempo transcurrido para las muestras de la URL dada.
- **Error %:** porcentaje de las peticiones con errores.
- **Rendimiento:** rendimiento medido en base a peticiones por segundo/minuto/hora.
- **Kb/sec:** rendimiento medido en Kilobytes por segundo.

Se puede observar que las pruebas se han realizado sin errores. Esto se deduce de la columna representativa del tanto por ciento de errores para cada una de las peticiones asociadas a cada conjunto de muestras. El rendimiento muestra que para una simulación de un usuario junto a un período de subida

## Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

de un segundo el servidor es capaz de aceptar una media de 2,4 peticiones por segundo. La latencia (entendida como el tiempo de espera para la renderización de la página, el tiempo en obtener respuesta del servidor) para cada conjunto de pruebas no supera el valor de 697 milisegundos (representado por el eje “y” de la gráfica).

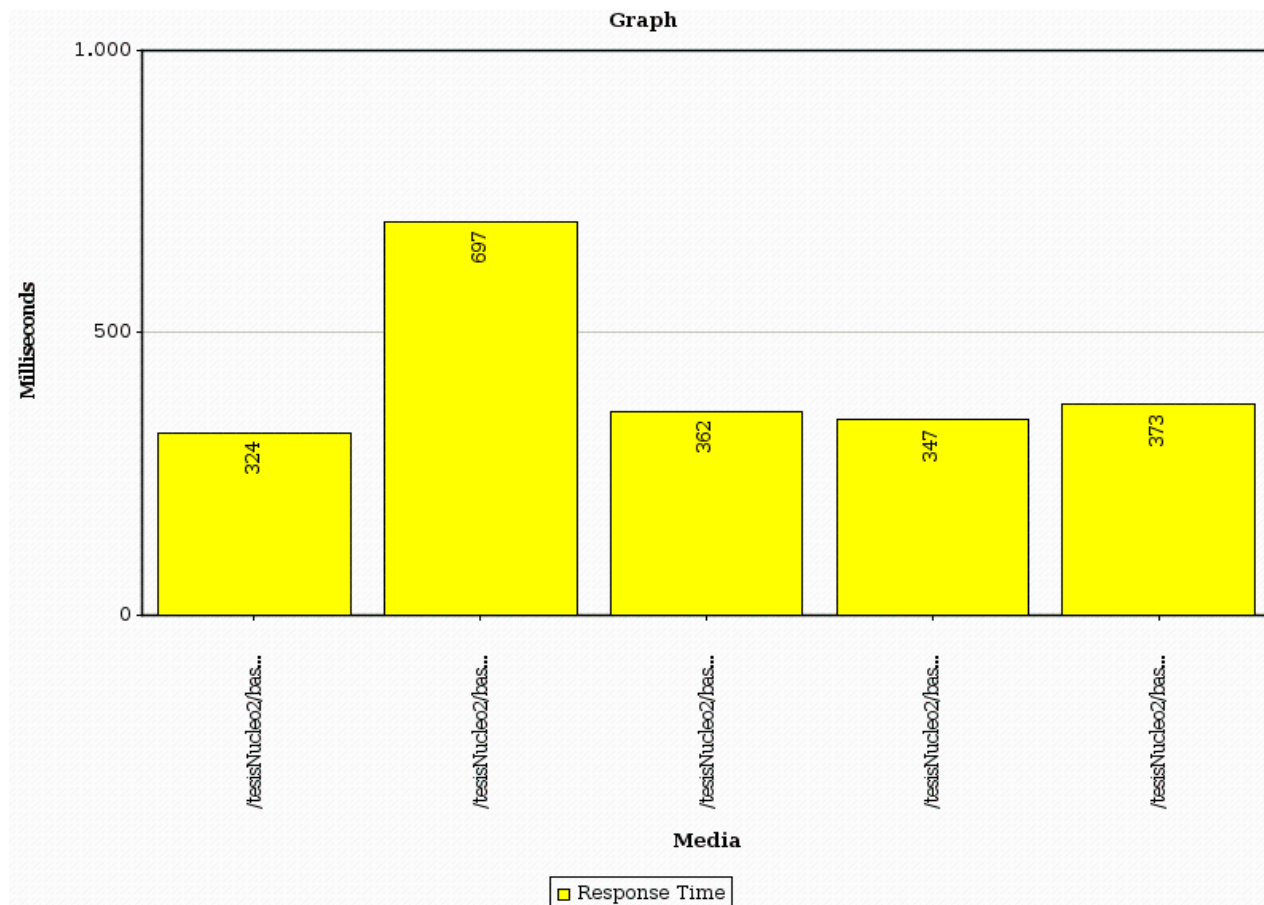


Figura 15 Gráfico de barra.

### Simulación de 50 usuarios con un período de subida de un segundo. Figura 6 del anexo.

En este caso se puede observar que las pruebas se han realizado sin errores. El rendimiento muestra que para una simulación de 50 usuarios junto a un período de subida de un segundo el servidor es capaz de aceptar una media de 7,2 peticiones por segundo. La latencia se ve aumentada hasta un valor de 6932 ms.

### Simulación de 80 usuarios con un período de subida de un segundo. Figura 7 del anexo.

De nuevo, no se ha obtenido errores puesto que se ha mantenido constante el tiempo transcurrido entre el lanzamiento de hilos. El rendimiento apenas se ha visto afectado. La latencia se ve aumentada hasta un valor de 11900 ms.

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### **Simulación de 100 usuarios con un período de subida de un segundo. Figura 8 del anexo.**

Se observa que comienzan a aparecer los primeros errores en un tanto por ciento considerable (10,80%) sobre el total de las peticiones realizadas por el global de las muestras. La latencia se ve aumentada hasta un valor de 12770 ms, lo cual se puede considerar como inaceptable.

Después de la realización de las pruebas para las muestras de 1, 50 y 80 usuarios la ocurrencia de errores se mantiene en 0.0%, lo que quiere decir que todas las peticiones hechas se ejecutan satisfactoriamente. Para una muestra mayor que 80, mayor será la probabilidad de ocurrencia de errores. Por tanto, se puede considerar que el límite permisible de procesamiento de peticiones concurrentes es de 80 usuarios con un período de subida de 1 segundo.

### **3.3.4 Pruebas de validación**

La validación se consigue cuando el software funciona de acuerdo con las expectativas razonables del cliente. Las expectativas razonables están definidas en la especificación de requisitos del software que describe todos los atributos del software visibles para el usuario. Estas pruebas se realizan a través de las técnicas alfa y beta. La prueba alfa se lleva a cabo, por un cliente, en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario y registrando los errores y los problemas de uso. Este tipo de prueba se lleva a cabo en un entorno controlado. La prueba beta se lleva a cabo por los usuarios finales del software en los lugares de trabajo de los clientes. A diferencia de la prueba alfa, el desarrollador no está presente normalmente. Así, la prueba beta es una aplicación en vivo del software en un entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas reales o imaginarios que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador. Como resultado de los problemas informados durante la prueba beta, el desarrollador del software lleva a cabo modificaciones y así prepara una versión del producto de software para toda la clase de clientes (Pressman, 2002). Para realizar las pruebas de aceptación a la solución se escogió la técnica alfa, tomando como referencia la descripción de requisitos y comprobando que el sistema cumple satisfactoriamente los requisitos del cliente, garantizando evaluar el grado de calidad del software.

### **Conclusiones parciales**

La caracterización de las técnicas de codificación empleadas en la implementación del sistema para la gestión de la propiedad intelectual, permitió una mejor organización y comprensión del código. Con la aplicación de técnicas y criterios para la validación de requisitos, se obtuvo un listado definitivo de requerimientos a implementar. La descripción de las pruebas realizadas para asegurar la calidad del software evidenció el correcto funcionamiento del sistema informático.

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### **Conclusiones generales**

A partir del desarrollo del presente trabajo se arriban a las siguientes conclusiones:

- ❖ Con el estudio del estado del arte se pudo identificar la no existencia de sistemas informáticos que incluyeran las particularidades del proceso de registro de la propiedad intelectual, basándose en las normativas establecidas a nivel de país relacionadas con el registro de obras y los procedimientos definidos por la Dirección de Comercialización de la Universidad.
- ❖ El estudio de los principales conceptos asociados al dominio de la problemática existente y el análisis realizado acerca de las principales herramientas, tecnologías y lenguajes seleccionados para el desarrollo de la solución informática, permitió la familiarización con el ambiente de desarrollo y la obtención de los conocimientos necesarios para poder utilizarlos en la construcción del sistema informático.
- ❖ Con la aplicación del proceso de desarrollo de software DAC, se logró obtener y especificar los requisitos, así como elaborar los artefactos propuestos y el diseño de la solución guiando adecuadamente el proceso de desarrollo del software.
- ❖ A partir del análisis y puesta en práctica de los patrones, estándares de diseño y estándares de código, se construyó el Módulo Propiedad Intelectual con las funcionalidades necesarias para facilitar la gestión de la propiedad intelectual de la Universidad de las Ciencias Informáticas.
- ❖ La aplicación de pruebas al software permitió detectar y corregir errores no identificados en la etapa de desarrollo; lográndose además, la aceptación de las funcionalidades implementadas por parte del cliente.



## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### **Recomendaciones**

- Incrementar el desarrollo de reportes relacionados con la información recogida en el proceso de registro de la propiedad intelectual con el objetivo de mostrar a través de listas, tablas y gráficos, los resúmenes que posibiliten al cliente o interesados la toma de decisiones.
- Desarrollar funcionalidades que permitan notificar vía correo electrónico a los solicitantes sobre el estado del proceso de las solicitudes que se realicen.

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### **Glosario de términos**

**AJAX:** Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas.

**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.

**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 que la creación de una obra literaria, artística, musical, científica o didáctica, esté publicada.

**Subversion:** sistema de control de versiones usado para que varios desarrolladores puedan trabajar en un mismo proyecto en forma más o menos ordenada.

**HTTP:** método de intercambio de información en la World Wide Web mediante el que se transfiere el contenido de las páginas web a un ordenador.

**PDF:** formato de almacenamiento de documentos digitales independiente de plataformas de software o hardware.

**FTP:** medio específico de conexión de un sitio web para cargar y descargar ficheros.

**DOM:** Modelo de Objetos de Documentos (DOM por sus siglas en inglés) es una interfaz de programación para los documentos HTML y XML. El DOM proporciona una representación del documento como un grupo estructurado de nodos y objetos que tienen propiedades y métodos. Esencialmente, comunica las páginas web para *scripts* o lenguajes de programación ( Mozilla Developer Network, 2005).

**GUUD:** nombre del marco de trabajo conformado con las iniciales de los departamentos del DIN: Gestión Universitaria, Universidad Digital y Gestión Documental.

## *Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

### **Referencias bibliográficas**

- Acuña, Diego, Schma, Rodolfo y Klein, Patricia. 2011.** *Una Plataforma Web para Gestionar los Derechos de Propiedad Intelectual.* Journal of Technology Management & Innovation. Chile : s.n., 2011.
- ALEGSA. 2010.** Diccionario de la informática. [En línea] 2010. [Citado el: 13 de 12 de 2013.] <http://www.alegsa.com.ar/Dic/uml.php>.
- Álvarez, Miguel Angel. 1999.** Desarrollo Web. [En línea] 1999. [Citado el: 16 de 12 de 2013.] <http://www.desarrolloweb.com>.
- Apache Software Foundation. 2013.** Apache JMeter. [En línea] Apache, 2013. [Citado el: 2014 de 4 de 12.] <http://jmeter.apache.org/>.
- Apache. 2012.** The Apache Software Foundation. [En línea] 2.0, 2012. [Citado el: 11 de 12 de 2013.] <http://www.apache.org/foundation/license-faq.html#GPL>.
- aprenderaprogramar.com. 2013.** Tutorial básico del programador web. *aprenderaprogramar.com*. [En línea] 2013. [Citado el: 15 de 11 de 2013.] URL: [http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=444:icuales-son-las-versiones-de-html-strict-transitional-frameset-diferencias-de-html-4-y-html-5-cu00706b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=](http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=444:icuales-son-las-versiones-de-html-strict-transitional-frameset-diferencias-de-html-4-y-html-5-cu00706b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=).
- Astudillo, Marcello Visconti y Hernán. 2003.** Departamento de Informática. [En línea] 2003. [Citado el: 8 de 4 de 2014.] <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>.
- Barchini, Graciela Elisa. 2005.** *Métodos "I + D" de la Informática.* Universidad Nacional de Santiago del Estero. Argentina : s.n., 2005.
- Canós, José H., Letelier, Patricio y Penadés, M<sup>a</sup> Carmen. 2008.** Desarrollo de Software con Metodologías Ágiles. [En línea] 2008. [http://noqualityinside.com.ar/nqi/nqifiles/XP\\_Agil.pdf](http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf).
- CENDA. 2010.** CENDA Centro Nacional de derecho de autor. [En línea] 2010. <http://www.cenda.cult.cu/php/loader.php>.
- CERLALC. 2003.** Derecho del Autor Regional. [En línea] 13 de 2 de 2003. [Citado el: 19 de 1 de 2014.] [http://www.cerlalc.org/derechoenlinea/dar/leyes\\_reglamentos/Cuba/resolucion\\_13.htm#top](http://www.cerlalc.org/derechoenlinea/dar/leyes_reglamentos/Cuba/resolucion_13.htm#top).
- Dirección para la Comercialización y los Negocios. 2013.** *Obras registradas.* La Habana : s.n., 2013.
- Instituto Tecnológico de Colima. 2010.** Instituto Tecnológico de Colima. [En línea] 2010. [Citado el: 10 de 4 de 2014.] [http://labredes.itcolima.edu.mx/fundamentosbd/sd\\_u1\\_6.htm](http://labredes.itcolima.edu.mx/fundamentosbd/sd_u1_6.htm).
- JQuery Foundation. 2013.** JQuery. [En línea] 2013. [Citado el: 14 de 1 de 2014.] <http://jquery.com/>.
- Larman, Craig. 2001.** UML y Patrones. *Introducción al análisis y diseño orientado a objetos y procesos unificado.* 2001.

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

- Microsoft. 2014.** Microsoft SQL Server. [En línea] 2014. <http://technet.microsoft.com/es-es/library/ms166026%28v=sql.90%29.aspx>.
- Mozilla Developer Network. 2013.** JavaScript. *Mozilla Developer Network*. [En línea] 2013. [Citado el: 11 de 2 de 2014.] <https://developer.mozilla.org/es/docs/JavaScript>.
- OMPI. 2008.** Organización Mundial de la Propiedad Mundial. [En línea] 2008. [Citado el: 19 de 2 de 2014.] <http://www.wipo.int/portal/es/>.
- Oracle. 2013.** Netbeans. [En línea] 2013. [Citado el: 13 de 12 de 2013.] [https://netbeans.org/community/releases/61/index\\_es.html](https://netbeans.org/community/releases/61/index_es.html).
- Pencil project. 2012.** Pencil project. [En línea] 2012. <http://pencil.evolus.vn/Features.html>.
- pgAdmin. 2010.** pgAdmin PostgreSQL. *Development*. [En línea] 2010. [Citado el: 24 de 2 de 2014.] <http://www.pgadmin.org/development/changelog.php>.
- PostgreSQL. 2009.** postgresQL-es. [En línea] 2009. [Citado el: 23 de 1 de 2014.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
- Pressman, Roger S. 2002.** *Un enfoque práctico*. Madrid : McGraw-Hill, 2002.
- Rodríguez Mompíe, Alexander y Sánchez Méndez, Alelí. 2013.** *Sistema Integral de Gestión Universitaria*. 2013.
- SafeCreative. 2008.** [En línea] 2008. <http://safecreative.org>.
- Sánchez Méndez, Alelí. 2013.** *Proceso de Desarrollo de Software. Metodología DAC*. La Habana : s.n., 2013.
- Scribd. 2013.** CodeIgniter. *Scribd*. [En línea] 2013. [Citado el: 24 de 11 de 2013.] <http://es.scribd.com/doc/68331716/CodeIgniter>.
- Semantic Copyright. 2009.** Semantic Copyright. [En línea] 2009. <http://semanticcopyright.org/>.
- Softonic Internacional S.A. 2012.** PHP. *Softonic*. [En línea] 2012. [Citado el: 12 de 12 de 2013.] <http://php.softonic.com/>.
- Sommerville, Ian. 2005.** *Ingeniería del software*. Séptima. Madrid : Pearson educación, 2005.
- Visual Paradigm para UML. 2010.** Visual Paradigm - UML. [En línea] 2010. [Citado el: 14 de 12 de 2013.] <http://www.visual-paradigm.com/aboutus/newsreleases/vpuml80.jsp>.

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

**Bibliografía**

- Acuña, Diego, Schma, Rodolfo y Klein, Patricia. 2011.** *Una Plataforma Web para Gestionar los Derechos de Propiedad Intelectual.* Journal of Technology Management & Innovation. Chile : s.n., 2011.
- ALEGSA. 2010.** Diccionario de la informática. [En línea] 2010. [Citado el: 13 de 12 de 2013.] <http://www.alegsa.com.ar/Dic/uml.php>.
- Álvarez, Miguel Angel. 1999.** Desarrollo Web. [En línea] 1999. [Citado el: 16 de 12 de 2013.] <http://www.desarrolloweb.com>.
- Álvarez, Miguel Ángel. 2001.** Teoría de la programación orientada a objetos. *desarrolloweb.com*. [En línea] 2001. <http://www.desarrolloweb.com/articulos/499.php>.
- Apache Software Foundation. 2013.** Apache JMeter. [En línea] Apache, 2013. [Citado el: 2014 de 4 de 12.] <http://jmeter.apache.org/>.
- Apache. 2012.** The Apache Software Foundation. [En línea] 2.0, 2012. [Citado el: 11 de 12 de 2013.] <http://www.apache.org/foundation/license-faq.html#GPL>.
- aprenderaprogramar.com. 2013.** Tutorial básico del programador web. *aprenderaprogramar.com*. [En línea] 2013. [Citado el: 15 de 11 de 2013.] URL: [http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=444:icuales-son-las-versiones-de-html-strict-transitional-frameset-diferencias-de-html-4-y-html-5-cu00706b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=](http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=444:icuales-son-las-versiones-de-html-strict-transitional-frameset-diferencias-de-html-4-y-html-5-cu00706b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=).
- Araluce, Rosalina Ramírez. 2013.** *Solución informática para el control del registro de la Propiedad Intelectual en la Universidad de las Ciencias Informáticas.* La Habana : s.n., 2013.
- Astudillo, Marcello Visconti y Hernán. 2003.** Departamento de Informática. [En línea] 2003. [Citado el: 8 de 4 de 2014.] <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>.
- Barchini, Graciela Elisa. 2005.** *Métodos "I + D" de la Informática.* Universidad Nacional de Santiago del Estero. Argentina : s.n., 2005.
- Canós, José H., Letelier, Patricio y Penadés, M<sup>a</sup> Carmen. 2008.** Desarrollo de Software con Metodologías Ágiles. [En línea] 2008. [http://noqualityinside.com.ar/nqi/nqifiles/XP\\_Agil.pdf](http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf).
- CENDA. 2010.** CENDA Centro Nacional de derecho de autor. [En línea] 2010. <http://www.cenda.cult.cu/php/loader.php>.
- CERLALC. 2003.** Derecho del Autor Regional. [En línea] 13 de 2 de 2003. [Citado el: 19 de 1 de 2014.] [http://www.cerlalc.org/derechoenlinea/dar/leyes\\_reglamentos/Cuba/resolucion\\_13.htm#top](http://www.cerlalc.org/derechoenlinea/dar/leyes_reglamentos/Cuba/resolucion_13.htm#top).
- Cole, Julio H. 2003.** *Propiedad Intelectual: Comentarios sobre algunas Tendencias recientes.* Universidad de Navarra. 2003. págs. 35-48.

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

- Correa, Carlos M. 2005.** *Propiedad Intelectual y Políticas de Desarrollo*. Argentina : Buenos Aires, 2005. pág. 489.
- desarrolloweb. 2009.** Modelo - Vista - Controlador en CodeIgniter. *desarrolloweb.com*. [En línea] 2009. [Citado el: 10 de 4 de 2014.] <http://www.desarrolloweb.com/articulos/modelo-vista-controlador-codeigniter.html>.
- Dirección para la Comercialización y los Negocios. 2013.** *Obras registradas*. La Habana : s.n., 2013.
- EcuRed. 2011.** Organización Mundial de la Propiedad Intelectual. *EcuRed*. [En línea] 2011. [Citado el: 28 de noviembre de 2013.] [http://www.ecured.cu/index.php/Organización\\_Mundial\\_de\\_la\\_Propiedad\\_Intelectual](http://www.ecured.cu/index.php/Organización_Mundial_de_la_Propiedad_Intelectual).
- Elmasri, Ramez y ShamKant B., Navathe. 2000.** *Sistemas de Bases de Datos*. s.l. : Addison Wesley, 2000.
- Guía de Ubuntu. 2008.** [En línea] 2008. [Citado el: 14 de 12 de 2013.] [http://www.guia-ubuntu.com/index.php?title=PgAdmin\\_III](http://www.guia-ubuntu.com/index.php?title=PgAdmin_III).
- Instituto Tecnológico de Colima. 2010.** Instituto Tecnológico de Colima. [En línea] 2010. [Citado el: 10 de 4 de 2014.] [http://labredes.itcolima.edu.mx/fundamentosbd/sd\\_u1\\_6.htm](http://labredes.itcolima.edu.mx/fundamentosbd/sd_u1_6.htm).
- Intellogist. 2009.** OCPI Web Sites. *Intellogist*. [En línea] 2009. [Citado el: 21 de 1 de 2014.] [http://www.intellogist.com/wiki/OCPI\\_Website\\_\(Cuba\)](http://www.intellogist.com/wiki/OCPI_Website_(Cuba)).
- JQuery Foundation. 2013.** JQuery. [En línea] 2013. [Citado el: 14 de 1 de 2014.] <http://jquery.com/>.
- Larman, Craig. 2001.** UML y Patrones. *Introducción al análisis y diseño orientado a objetos y procesos unificado*. 2001.
- Microsoft. 2014.** Developer network. [En línea] 2014. <http://msdn.microsoft.com/es-es/library/aa577691.aspx>.
- . 2014.** Microsoft SQL Server. [En línea] 2014. <http://technet.microsoft.com/es-es/library/ms166026%28v=sql.90%29.aspx>.
- Mozilla Developer Network. 2013.** JavaScript. *Mozilla Developer Network*. [En línea] 2013. [Citado el: 11 de 2 de 2014.] <https://developer.mozilla.org/es/docs/JavaScript>.
- . 2005.** Mozilla Developer Network. [En línea] 2005. [Citado el: 23 de 1 de 2014.] [https://developer.mozilla.org/en-US/docs/DOM/DOM\\_Reference](https://developer.mozilla.org/en-US/docs/DOM/DOM_Reference).
- OCPI. 2006.** Oficina Cuba de la Propiedad Intelectual. [En línea] 2006. [Citado el: 10 de 12 de 2013.] <http://www.ocpi.cu/>.
- OMPI. 2008.** Organización Mundial de la Propiedad Intelectual. [En línea] 2008. [Citado el: 19 de 2 de 2014.] <http://www.wipo.int/portal/es/>.

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

- Oracle. 2013.** Netbeans. [En línea] 2013. [Citado el: 13 de 12 de 2013.] [https://netbeans.org/community/releases/61/index\\_es.html](https://netbeans.org/community/releases/61/index_es.html).
- Pencil project. 2012.** Pencil project. [En línea] 2012. <http://pencil.evolus.vn/Features.html>.
- Pérez, Sergio Ovalle. 2011.** Maestros del Web. [En línea] 23 de agosto de 2011. [Citado el: 29 de noviembre de 2013.] <http://www.maestrosdelweb.com/editorial/guia-propiedad-intelectual/>.
- pgAdmin. 2010.** pgAdmin PostgreSQL. *Development*. [En línea] 2010. [Citado el: 24 de 2 de 2014.] <http://www.pgadmin.org/development/changelog.php>.
- PostgreSQL. 2009.** postgresql-es. [En línea] 2009. [Citado el: 23 de 1 de 2014.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
- Pressman, Roger S. 2002.** *Un enfoque práctico*. Madrid : McGraw-Hill, 2002.
- Programación en java. 2011.** Programación en java. [En línea] 2011. <http://programarjava.wordpress.com/2011/12/13/programacion-orientada-a-eventos/>.
- Registro de Obras. [En línea] [Citado el: 29 de noviembre de 2013.] <http://www.cenda.cult.cu/php/loader.php>.
- Revista chilena de derecho. 2009.** *Pontificia Universidad Católica de Chile. Facultad de Derecho*. Santiago de Chile : s.n., 2009. 0718-3437.
- Rodríguez Mompíe, Alexander y Sánchez Méndez, Alelí. 2013.** *Sistema Integral de Gestión Universitaria*. 2013.
- SafeCreative. 2008.** [En línea] 2008. <http://safecreative.org>.
- Sánchez Méndez, Alelí. 2013.** *Proceso de Desarrollo de Software. Metodología DAC*. La Habana : s.n., 2013.
- Scribd. 2013.** Codelgniter. *Scribd*. [En línea] 2013. [Citado el: 24 de 11 de 2013.] <http://es.scribd.com/doc/68331716/Codelgniter>.
- Semantic Copyright. 2009.** Semantic Copyrigh. [En línea] 2009. <http://semanticcopyright.org/>.
- Sherwood, Robert M. 1992.** *Propiedad Intelectual y Desarrollo Económico*. Buenos Aires : Heliasta, 1992. pág. 250.
- Softonic Internacional S.A. 2012.** PHP. *Softonic*. [En línea] 2012. [Citado el: 12 de 12 de 2013.] <http://php.softonic.com/>.
- Soler, Otto Batista, Pérez Valdéz, Ignacio y Bonal Cáceres, Rolando. 2008.** *Infraestructura productiva manual de procedimientos*. 2008.
- Sommerville, Ian. 2005.** *Ingeniería del software*. Septima. Madrid : Pearson educación, 2005.
- Valdéz, Damián Pérez. 1997.** Maestros Web. [En línea] 3 de Julio de 1997. [Citado el: 13 de 12 de 2013.] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

**Vidal, Yanio Garcia. 2007.** *Infraestructura Productiva.* 2007.

**Visual Paradigm para UML. 2010.** Visual Paradigm - UML. [En línea] 2010. [Citado el: 14 de 12 de 2013.] <http://www.visual-paradigm.com/aboutus/newsreleases/vpuml80.jsp>.

**XING. 2014.** Definiciones. *XING.* [En línea] 2014. <http://www.xing.com/net/intangible/general-144223/definiciones-intangible-13560766>.



# Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

## Anexos

### Anexo 1 Pruebas de integración

Módulo al que se integra	Seguridad
<b>Condiciones de ejecución:</b> el módulo Seguridad debe tener los datos en la base de datos central y debe existir conexión con la misma.	
<b>Descripción de la prueba:</b> comprobar que el módulo Propiedad Intelectual es capaz de realizar el acceso a las funcionalidades y seguridad a partir de la información gestionada en el módulo Seguridad.	
<b>Pasos de ejecución:</b> el módulo Seguridad introduce en la base de datos central los datos y el módulo Propiedad Intelectual consulta estos datos y define la seguridad del módulo.	
<b>Resultados esperados:</b> los usuarios tienen acceso a las funcionalidades definidas de acuerdo a su rol y responsabilidades.	
<b>Evaluación:</b> prueba satisfactoria.	

Tabla 9 Prueba de integración del módulo Propiedad Intelectual con Seguridad.

Módulo al que se integra	Personal
<b>Condiciones de ejecución:</b> el módulo Personal haya introducido los datos en la base de datos central y exista conexión con la misma.	
<b>Descripción de la prueba:</b> comprobar que el módulo Propiedad Intelectual es capaz de confeccionar la solicitud con información gestionada por el módulo de Personal.	
<b>Pasos de ejecución:</b> el módulo Personal introduce en la base de datos central los datos y el módulo Propiedad Intelectual consulta estos datos y define la seguridad del módulo.	
<b>Resultados esperados:</b> los usuarios tienen acceso a las funcionalidades definidas de acuerdo a su rol y responsabilidades.	
<b>Evaluación:</b> prueba satisfactoria.	

Tabla 10 Prueba de integración del módulo Propiedad Intelectual con Personal.

## Anexo 2 Prueba de unidad

<b>Prueba estructural de caja blanca</b>	<b>Código caso de prueba:</b>
	Obtener tipo de obra dado ID
<b>Probador:</b>	Yusel Carlos Carralero Calzadilla
<b>Tipo de dato esperado:</b>	Integer
<b>Código al que se aplica:</b>	
<pre>public function obtenerTipoObraDadoId(\$id) {     \$id_tipo_obra = array('id_tipo_obra' =&gt; \$id);     \$tipo_obra = \$this-&gt;_ci-&gt;tb_ntipo_obra_md1-&gt;obtenerDadoId(\$id_tipo_obra);     if (!empty(\$tipo_obra))         return \$tipo_obra;     return FALSE; }</pre>	
<b>Función de evaluación:</b>	
<pre>public function prueba(){     \$dato=53;     \$datos= new stdClass();     \$datos-&gt;id_tipo_obra=53;     \$datos-&gt;nombre_tipo_obra='Audiovisuales';     \$datos-&gt;descripcion= 'descrip';     \$datos-&gt;fecha_registro='2014-05-05 22:19:08.711685';     \$datos-&gt;activo = 't';      echo \$this-&gt;_ci-&gt;unit-&gt;run(\$this-&gt;obtenerTipoObraDadoId(\$dato),\$datos,         "Prueba:'Obtener obra'"); }</pre>	

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

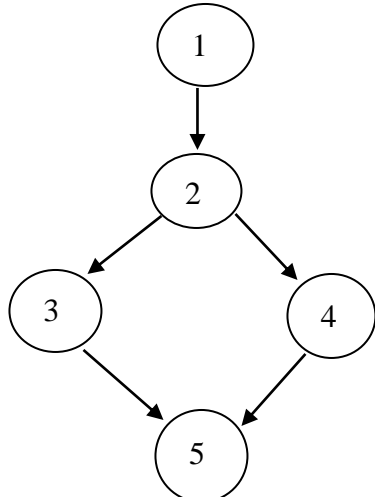
<p><b>Complejidad ciclomática:</b>  <math>V(G) = (A-N) + 2 = (5-5) + 2 = 2</math></p> <p><b>Caminos independientes:</b></p> <ol style="list-style-type: none"> <li>1-2-3-5</li> <li>1-2-3-4</li> </ol>	<p><b>Representación del grafo:</b></p>  <pre> graph TD     1((1)) --&gt; 2((2))     2((2)) --&gt; 3((3))     2((2)) --&gt; 4((4))     3((3)) --&gt; 5((5))     4((4)) --&gt; 5((5))     </pre>												
<p><b>Resultado de la prueba:</b></p> <table border="1" data-bbox="162 913 1518 1102"> <tr> <td>Test Name</td> <td>Prueba:'Obtener obra'</td> </tr> <tr> <td>Test Datatype</td> <td>Object</td> </tr> <tr> <td>Expected Datatype</td> <td>Object</td> </tr> <tr> <td>Result</td> <td>Passed</td> </tr> <tr> <td>File Name</td> <td>/var/www/tesisNucleo3/base/application/produccion/propiedad_intelectual/libraries/tipo_obra_lib.php</td> </tr> <tr> <td>Line Number</td> <td>98</td> </tr> </table>		Test Name	Prueba:'Obtener obra'	Test Datatype	Object	Expected Datatype	Object	Result	Passed	File Name	/var/www/tesisNucleo3/base/application/produccion/propiedad_intelectual/libraries/tipo_obra_lib.php	Line Number	98
Test Name	Prueba:'Obtener obra'												
Test Datatype	Object												
Expected Datatype	Object												
Result	Passed												
File Name	/var/www/tesisNucleo3/base/application/produccion/propiedad_intelectual/libraries/tipo_obra_lib.php												
Line Number	98												
<p><b>Evaluación del caso de prueba:</b></p>	<p>Satisfactoria</p>												

Tabla 11 Prueba de unidad Obtener tipo de obra.

<p><b>Prueba estructural de caja blanca</b></p>	<p><b>Código caso de prueba:</b>                  Obtener cantidad de tipos de obras</p>
<p><b>Probador:</b></p>	<p>Yusel Carlos Carralero Calzadilla</p>
<p><b>Tipo de dato esperado:</b></p>	<p>Integer</p>
<p><b>Código al que se aplica:</b></p> <pre> public function obtenerCantidadTipo0bra(\$inicio=null, \$limite=null,     \$elementoOrdenar='', \$dirOrdenar='asc', \$params = array()) {     return count(\$this-&gt;_ci-&gt;tb_ntipo_obra_md1-&gt;obtenerTipoobraPorPagina(\$inicio,         \$limite, 'activo', 'desc', \$params)); }                 </pre>	
<p><b>Función de evaluación:</b></p> <pre> public function prueba(){     echo \$this-&gt;_ci-&gt;unit-&gt;run(\$this-&gt;obtenerCantidadTipo0bra(),5,         "Prueba:'Cantidad de tipo obra"); }                 </pre>	

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*


<p><b>Complejidad ciclomática:</b>  <math>V(G) = (A-N) + 2 = (0-1) + 2 = 1</math></p> <p><b>Caminos independientes:</b>          1. 1</p>	<p><b>Representación del grafo:</b></p> 												
<p><b>Resultado de la prueba:</b></p> <table border="1"> <tr> <td>Test Name</td> <td>Prueba:'Cantidad de tipo obra'</td> </tr> <tr> <td>Test Datatype</td> <td>Integer</td> </tr> <tr> <td>Expected Datatype</td> <td>Integer</td> </tr> <tr> <td>Result</td> <td>Passed</td> </tr> <tr> <td>File Name</td> <td>/var/www/tesisNucleo3/base/application/produccion/propiedad_intelectual/libraries/tipo_obra_lib.php</td> </tr> <tr> <td>Line Number</td> <td>28</td> </tr> </table>		Test Name	Prueba:'Cantidad de tipo obra'	Test Datatype	Integer	Expected Datatype	Integer	Result	Passed	File Name	/var/www/tesisNucleo3/base/application/produccion/propiedad_intelectual/libraries/tipo_obra_lib.php	Line Number	28
Test Name	Prueba:'Cantidad de tipo obra'												
Test Datatype	Integer												
Expected Datatype	Integer												
Result	Passed												
File Name	/var/www/tesisNucleo3/base/application/produccion/propiedad_intelectual/libraries/tipo_obra_lib.php												
Line Number	28												
<p><b>Evaluación del caso de prueba:</b></p>	<p>Satisfactoria</p>												

Tabla 12 Prueba de unidad Obtener cantidad de tipos de obra.

# Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

## Anexo 3 Pruebas de carga y stress

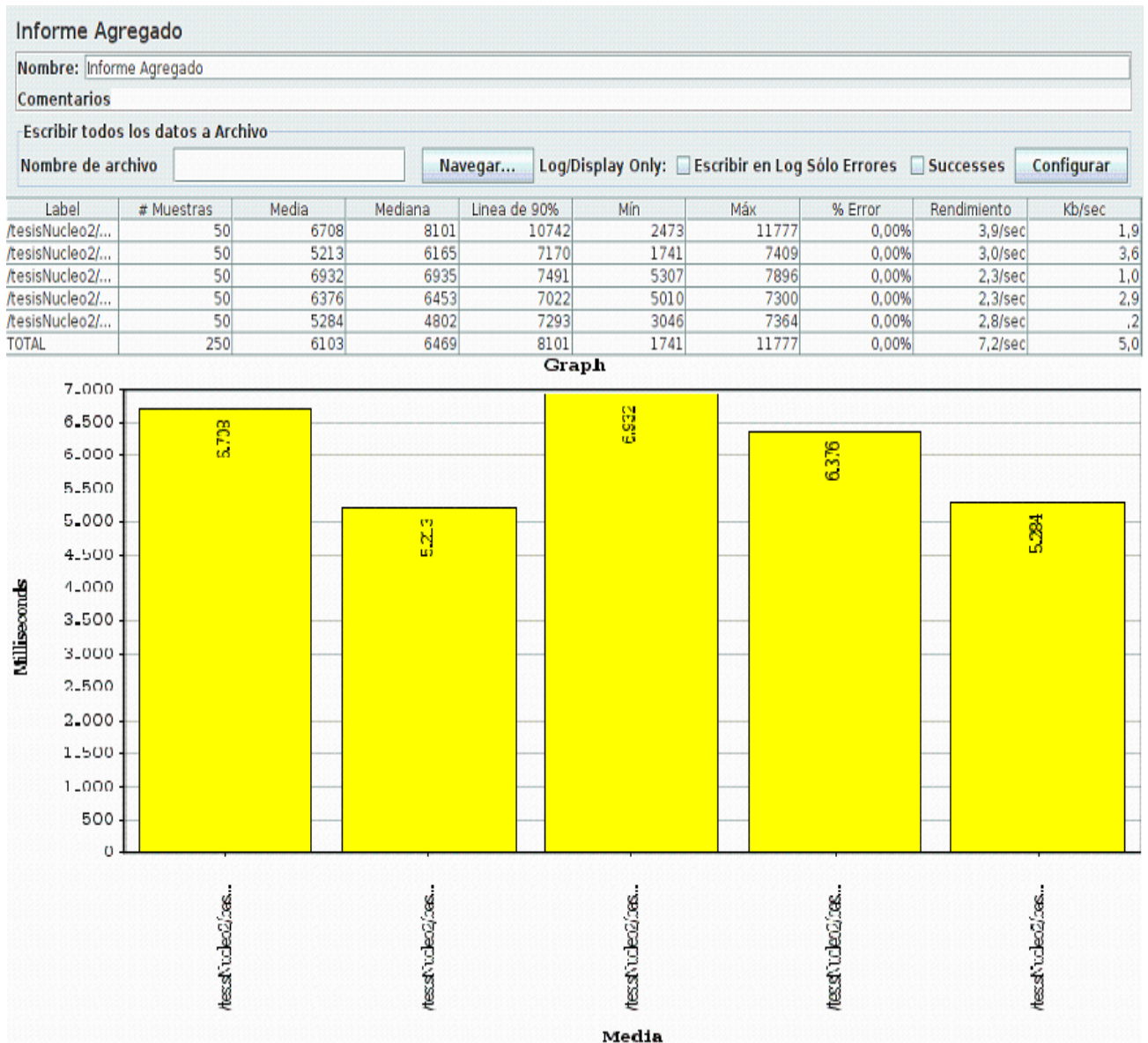


Figura 16 Pruebas de carga y stress resultado para 50 muestras.

Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.

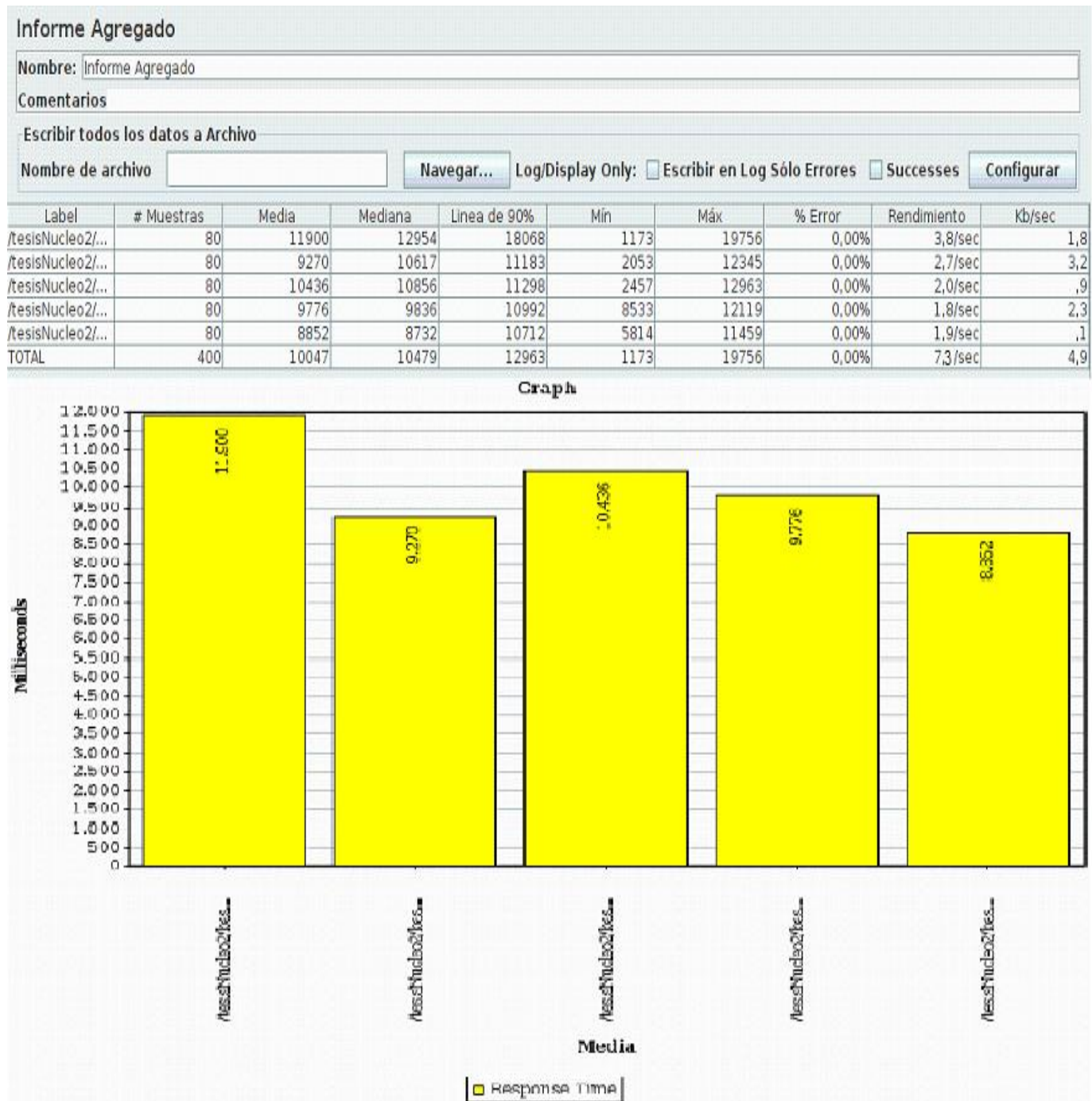


Figura 17 Pruebas de carga y stress resultado para 80 muestras.

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*



**Figura 18 Pruebas de carga y stress resultado para 100 muestras.**



*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

**Anexo 4 Diseño de casos de prueba**

Escenario	Descripción	Variable 1. Nombre del proyecto	Variable 2. Nombre del producto	Variable 3. Áreas
EC 1.1 Insertar datos correctamente.	Este escenario permite crear una solicitud insertando los datos correctamente.	V	V	V
		Nova	Nova	Facultad 1
EC 1.2 Insertar datos repetidos.	Mediante este escenario se pretende insertar una solicitud utilizando datos existentes en el sistema.	V	V	V
		Nova	Nova	Facultad 1
EC 1.3 Insertar datos incompletos.	Mediante este escenario se dejan campos requeridos sin llenar para crear una solicitud.	V	V	V
		Vacío	Vacío	Facultad 2
EC 1.4 Insertar datos incorrectos.	Mediante este escenario se introducen datos incorrectos para crear una solicitud.	V	V	V
		El usuario introduce más de 30 caracteres	CedruX	Centro de soporte
		I	V	V
		%ScaWeb	Buscador	Facultad 3
		V	V	V
		MATELEC	Catálogo Electrónico Ecoso	Facultad 2
		I	V	V
		A	Drupal	Facultad 5
EC 1.5 Cancelar	Mediante este			



*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

operación.	escenario se cancela la operación de crear solicitud.			
------------	---	--	--	--

<b>Variable 4. Clasificación</b>	<b>Variable 5. Resumen</b>	<b>Variable 6. Versión</b>	<b>Variable 7. Tipo de obra</b>	<b>Variable 8. Tipología</b>
V	N/A	V	V	V
Sistema operativo.	Nova brinda gran facilidad de uso y las aplicaciones orientadas al usuario final. Las principales aplicaciones que trae Nova son: El navegador web Mozilla Firefox, suite ofimática Open Office, etc.	3	Software	Sistema operativo
V	N/A	V	V	V
Sistema operativo.	Nova brinda gran facilidad de uso y las aplicaciones orientadas al usuario final. Las principales aplicaciones que trae Nova son: El navegador web Mozilla Firefox, suite ofimática Open Office, etc.	3	Software	Sistema operativo
V	N/A	V	V	V
Educativo	Portal educativo para la enseñanza media.	1	Software	Portal web
V	N/A	V	V	V
Desarrollo de Software.		3	Software	Módulo

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

V	N/A	V	V	V
Gestión de contenido		2	Software	Web
V	N/A	V	V	V
Desarrollo de Software		1	Software	Multimedia
V	N/A	I	V	V
Software		tres	Software	Multimedia

Variable 9. Entidad de registro	Variable 10. Organismo titular	Variable 11. Autores	Respuesta del sistema	Flujo central
V	V	V	El sistema muestra el mensaje: El elemento ha sido creado satisfactoriamente.	El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona el módulo 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.
OCPI	UCI	Yusel Carlos Carralero Calzadilla, Yasiel Pérez Vera		

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

V	V	V	El sistema muestra el mensaje: El elemento ya existe.	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.
OCPI	UCI	Yusel Carlos Carralero Calzadilla, Yasiel Pérez Vera		
V	V	V	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 Propiedad Intelectual y luego en el menú lateral, en la agrupación funcional Solicitud selecciona la funcionalidad Solicitud.
CENDA	UCI	Yurieski Madrigal Vilches		

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

				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 Aceptar.
V	V	V	El sistema muestra el mensaje de error: "Entre al menos 2 caracteres" sobre el campo requerido.	El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona el módulo Propiedad Intelectual y luego en el menú lateral, en la agrupación funcional Solicitud selecciona la funcionalidad Solicitud.
CENDA	UCI	Orlando Martínez Batista		
V	V	V	El sistema no permite seguir escribiendo.	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 Aceptar.
CENDA	UCI	Julio Edel Morfe Díaz		
V	V	V	El sistema muestra en rojo el mensaje: "Solo entre letras y espacios" sobre el campo requerido.	El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona el módulo Propiedad Intelectual y luego en el menú lateral, en la agrupación funcional Solicitud selecciona la funcionalidad Solicitud.
CENDA	ETECSA	José Pérez Dril		
V	V	V	El sistema muestra en rojo el mensaje: "Entre solo números."	El usuario una vez autenticado en el Sistema de Gestión Universitaria, selecciona el módulo Propiedad Intelectual y luego en el menú lateral, en la agrupación funcional Solicitud selecciona la funcionalidad Solicitud.
CENDA	UCI	Julio Fernández López		
			El sistema muestra	El usuario una vez

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

			<p>el mensaje de confirmación ¿Está seguro de realizar la acción?</p>	<p>autenticado en el Sistema de Gestión Universitaria, selecciona el módulo 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 y presiona el botón Cancelar.</p>
--	--	--	---	---

**Tabla 13** Diseño del caso de prueba crear solicitud.

Escenario	Descripción	Nombre rol	Descripción	Habilitado	Respuesta del sistema	Flujo central
EC 1.1 Insertar	Mediante este	V	NA	NA	Muestra un	El requisito comienza cuando

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

todos los datos correctamente.	escenario el usuario inserta correctamente los datos para la creación de un rol.	Diseñador de Base de Datos.	El diseñador de Base de Datos posee un papel importante en cualquier desarrollo de aplicaciones web, tiene como responsabilidad elaborar el DER y el MER, de la Base de Dato de la aplicación.	Si	mensaje diciendo: "El elemento ha sido creado satisfactoriamente".	el administrador del sistema selecciona del módulo Propiedad Intelectual del subsistema Producción, la opción Rol productivo del menú de funcionalidades Configuración. El administrador del sistema selecciona la opción Crear en el área de íconos flotantes e incorpora los siguientes datos: - Nombre rol (obligatorio) - Descripción - Habilitado Una vez especificado los datos el administrador selecciona la opción Aceptar, mostrándose el mensaje: "El elemento ha sido creado satisfactoriamente". Guardándose los datos de la entidad creada y mostrándose el listado de roles. Se muestra en el área de iconos flotantes el icono Listar.
EC 1.2 Insertar los datos incorrectamente.	En este escenario el usuario inserta al menos un carácter no admitido por la aplicación.	I Programador %	NA La persona que desarrolla este rol se encarga de implementar los métodos	NA Si	Muestra un mensaje diciendo: "Entre solo letras, espacios".	El requisito comienza cuando el administrador del sistema selecciona del módulo Propiedad Intelectual del subsistema Producción, la opción rol productivo del menú de funcionalidades Configuración.

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

			de acuerdo a los requisitos funcionales del sistema.			El administrador del sistema selecciona la opción Crear en el área de íconos flotantes e incorpora los siguientes datos: - Nombre rol (obligatorio) - Descripción - Habilitado Una vez especificado los datos el administrador selecciona la opción Aceptar.
EC 1.3 Insertar datos repetidos.	Mediante este escenario se pretende insertar un rol utilizando datos existentes en el sistema.	V	NA	V	El sistema muestra el mensaje: "El elemento ya existe".	El requisito comienza cuando el administrador del sistema selecciona del módulo Propiedad Intelectual del subsistema Producción, la opción rol productivo del menú de funcionalidades Configuración. El administrador del sistema selecciona la opción Crear en el área de íconos flotantes e incorpora los siguientes datos: - Nombre rol (obligatorio) - Descripción - Habilitado Una vez especificado los datos el administrador selecciona la opción Aceptar.
		Analista		Si		
EC 1.4 Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para crear una	I	NA	V	El sistema muestra encima del componente un mensaje en color rojo	El requisito comienza cuando el administrador del sistema selecciona del módulo Propiedad Intelectual del subsistema Producción, la opción rol productivo del
		Vacío		Activo		

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

	variable.				indicando: "Campo requerido".	menú de funcionalidades Configuración. El administrador del sistema selecciona la opción Crear en el área de íconos flotantes e incorpora los siguientes datos: - Nombre rol (obligatorio) - Descripción - Habilitado Una vez especificado los datos el administrador selecciona la opción Aceptar.
EC 1.5 Cancelar operación.	Mediante este escenario se cancela la creación de una variable	NA	NA	NA	El sistema muestra el mensaje de confirmación ¿Está seguro de realizar la acción?	El requisito comienza cuando el administrador del sistema selecciona del módulo Propiedad Intelectual del subsistema Producción, la opción rol productivo del menú de funcionalidades Configuración. El administrador del sistema selecciona la opción Crear en el área de íconos flotantes. El usuario llena o no los datos y presiona el botón Cancelar.

**Tabla 14 Diseño del caso de prueba registrar rol productivo.**



*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

Escenario	Descripción	Nombre rol	Descripción	Habilitado	Respuesta del sistema	Flujo central
EC 1.1 Modificar los datos correctamente	En este escenario el usuario modifica el rol introduciendo correctamente los parámetros.	V Programador	NA	NA Si	Muestra un mensaje diciendo "El elemento se ha modificado correctamente".	El requisito comienza cuando el administrador del sistema selecciona el módulo Propiedad Intelectual del subsistema Producción, la opción rol productivo del menú de funcionalidades Configuración. El administrador del sistema selecciona la opción Modificar en el listado de roles mostrándose los datos correspondiente a los roles (todos los datos pueden modificarse): <ul style="list-style-type: none"> <li>- Nombre rol (obligatorio)</li> <li>- Descripción</li> <li>- Habilitado</li> </ul> Una vez especificado los datos el administrador selecciona la opción Aceptar, mostrándose el mensaje: "El elemento ha sido modificado satisfactoriamente". Guardándose los datos del rol modificado. Se muestra en el área de iconos flotantes el icono

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

						Listar.
EC 1.2 Entrar los datos incorrectamente.	En este escenario el usuario inserta al menos un carácter que no admite la aplicación.	I	NA	NA	Muestra un mensaje diciendo "Entre solo letras y espacios".	El requisito comienza cuando el administrador del sistema selecciona del módulo Propiedad Intelectual del subsistema Producción, la opción rol productivo del menú de funcionalidades Configuración. El administrador del sistema selecciona la opción Modificar en el listado de roles mostrándose los datos correspondiente a las entidades de registro (todos los datos pueden modificarse): <ul style="list-style-type: none"> <li>- Nombre rol (obligatorio)</li> <li>- Descripción</li> <li>- Habilitado</li> </ul> Una vez especificado los
		Probador1		No		

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

						datos el administrador selecciona la opción Aceptar, mostrándose el mensaje: "Entre solo letras y espacios".
EC 1.3 Insertar elementos repetidos	Mediante este escenario se introducen datos para modificar un rol que ya existe en el sistema.	V	NA	NA	El sistema muestra en forma de ventana el mensaje de error indicando: "El elemento ya existe".	El requisito comienza cuando el administrador del sistema selecciona del módulo Propiedad Intelectual del subsistema Producción, la opción rol productivo del menú de funcionalidades Configuración. El administrador del sistema selecciona la opción Modificar en el listado de roles mostrándose los datos correspondiente a las entidades de registro (todos los datos pueden modificarse): - Nombre rol (obligatorio) - Descripción - Habilitado Una vez especificado los datos el administrador selecciona la opción
		Programador				

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

						Aceptar.
EC 1.4 Insertar datos incompletos.	Mediante este escenario no se introducen todos los datos para modificar un rol.	I  Vacío	NA  	NA  	El sistema muestra encima del componente un mensaje en color rojo indicando: "Campo requerido".	El requisito comienza cuando el administrador del sistema selecciona del módulo Propiedad Intelectual del subsistema Producción, la opción rol productivo del menú de funcionalidades Configuración. El administrador del sistema selecciona la opción Modificar en el listado de roles mostrándose los datos correspondiente a las entidades de registro (todos los datos pueden modificarse): - Nombre rol (obligatorio) - Descripción - Habilitado Una vez especificado los datos el administrador selecciona la opción Aceptar.
EC 1.5	Mediante este	NA	NA	NA	El sistema	El requisito comienza

*Solución informática para la gestión de la propiedad intelectual en la Universidad de las Ciencias Informáticas.*

<p>Cancelar operación.</p>	<p>escenario se cancela la creación de un rol productivo.</p>				<p>muestra el mensaje de confirmación ¿Está seguro de realizar la acción?</p>	<p>cuando el administrador del sistema selecciona del módulo Propiedad Intelectual del subsistema Producción, la opción rol productivo del menú de funcionalidades Configuración. El administrador del sistema selecciona la opción Modificar en el listado de roles mostrándose los datos correspondiente a las entidades de registro (todos los datos pueden modificarse):</p> <ul style="list-style-type: none"> <li>- Nombre rol (obligatorio)</li> <li>- Descripción</li> <li>- Habilitado</li> </ul> <p>Una vez especificado los datos el administrador selecciona la opción Aceptar.</p>
----------------------------	---	--	--	--	---	---

**Tabla 15** Diseño de caso de prueba modificar rol.