

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 5

CENTRO DE INFORMÁTICA INDUSTRIAL



Universidad de las Ciencias
Informáticas

Sistema para la gestión del registro de la Propiedad Intelectual para el Centro de Informática Industrial

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

Autor: Carlos Silva Alvarez

Tutor (es): Ing. Sonia Beatriz Franco Kile

Ing. Yaima Antunez Ojeda

La Habana, Junio 2014

“Año 56 del Triunfo de la Revolución”



“Es muy justo luchar por eso, y por eso debemos emplear todas nuestras energías, todos nuestros esfuerzos, todo nuestro tiempo para poder decir en la voz de millones o de cientos o de miles de millones: ¡Vale la pena haber nacido! ¡Vale la pena haber vivido!”

Fidel Castro Ruz

AGRADECIMIENTOS

Después de cinco años incansables de estudio se hace difícil recordar todas aquellas personas que merecen ser agradecidas por aportar de una forma u otra elementos positivos a tu vida diaria. ¡Hagamos un intento!

A mi mamá, mi papá y mi hermana; ustedes son mi razón de ser, la luz que ilumina mis caminos. Gracias por existir...

A mi familia y en especial a mis tías, mis tíos, mi abuelo y mis abuelas queridas.

A Betty y Yaima; por haber confiado en mí para realizar esta investigación. Gracias por abrir sus brazos y tenerme como un amigo. ¡Las quiero...!

Agradecer además, a Lisandra; mi amiga de 5 años de carrera, compañera de estudio. Gracias por los momentos bonitos que tuvimos...

A Roberto, Katy y Yunior, por los buenos momentos que hemos compartido

A todos los profesores que aportaron su granito de arena en mi desarrollo como profesional.

A los profes de Idioma Extranjero: Zoraida, Zaida y Fermin; thank you so much for supporting me.

A todas las amistades que la UCI me dio la oportunidad de tener; Julio, Roger, Tanya, Henry, Luis Manuel, Yidian, Yoandry, Yamila, Ernesto Alejandro.

A mi gente del 88-204; Julio, Polo, Pori, el negro, Ernesto, Hubert, Henry y Gladys por aguantarme en estos 5 años.

Al Director de Contratación y Servicios de la universidad, por toda la información brindada.

DEDICATORIA

*A mis padres y mi hermana, que siempre son mi fuerza,
mi apoyo, mis mejores críticos y mi guía en todo lo
que he hecho y haré.*

*A mi familia, por tenerme siempre presente
y estar ahí cuando la necesito.*

*A mis amistades de la universidad, por todo lo
que vivimos y compartimos en estos 5 años de vida.*

DECLARACIÓN JURADA DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 5 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Carlos Silva Alvarez

Ing. Sonia Beatriz Franco Kile

Ing. Yaima Antunez Ojeda

RESUMEN

El progreso vertiginoso que ha tenido el desarrollo de software en el país ha llevado a los centros de desarrollo a buscar alternativas y mecanismos para dar protección a cada uno de sus productos. La Universidad de las Ciencias Informáticas como ejemplo primordial de dichos avances, le da gran importancia al registro de los productos que se implementan en sus centros. El Centro de Informática Industrial (CEDIN) de dicha universidad, se dedica al desarrollo de productos de alta calidad que permiten la optimización del trabajo en los sectores industriales. Actualmente el proceso se hace manual lo que atenta contra la agilidad del mismo y la veracidad de la información que se tiene. Por esta razón el objetivo de la investigación es desarrollar una aplicación informática para gestionar el proceso de registro de la propiedad intelectual del CEDIN. Como resultado se obtuvo una aplicación que constituye una alternativa fiable pues aporta al centro una aplicación confiable y multiplataforma para llevar a cabo el proceso de registro del software.

Palabras clave: propiedad intelectual, protección, software.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	4
1.1. PROPIEDAD INTELECTUAL. DEFINICIONES Y CATEGORÍAS.....	4
1.1.1. <i>Propiedad Industrial</i>	5
1.1.2. <i>Derecho de Autor</i>	6
1.2. REGISTRO DE LA PROPIEDAD INTELECTUAL EN CUBA.....	6
1.3. SISTEMAS DE GESTIÓN DE LA PROPIEDAD INTELECTUAL.....	7
1.4. METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	9
1.4.1. <i>Metodologías pesadas</i>	9
1.4.2. <i>Metodologías ágiles</i>	10
1.4.3. <i>Metodología de desarrollo de software utilizada</i>	11
1.5. HERRAMIENTAS CASE.....	12
1.6. LENGUAJE DE PROGRAMACIÓN.....	13
1.6.1. <i>Lenguajes de programación en el servidor</i>	13
1.6.2. <i>Lenguajes de programación para servirle al cliente</i>	13
1.7. SISTEMAS DE GESTIÓN DE BASE DE DATOS.....	15
1.7.1. <i>PostgreSQL</i>	15
1.7.2. <i>MySQL</i>	15
1.7.3. <i>Sistema de Gestión de Bases de Datos a utilizar</i>	16
1.8. ENTORNO DE DESARROLLO INTEGRADO.....	16
1.9. FRAMEWORK.....	17
1.10. HERRAMIENTAS, METODOLOGÍAS Y LENGUAJES A UTILIZAR.....	17
CONSIDERACIONES PARCIALES.....	17
CAPÍTULO 2: DESCRIPCIÓN Y DISEÑO DE LA SOLUCIÓN PROPUESTA.....	19
2.1. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.....	19
2.2. PERSONAL RELACIONADO CON EL SISTEMA.....	19
2.3. REQUISITOS DEL SISTEMA.....	20
2.3.1. <i>Requisitos funcionales</i>	20
2.3.2. <i>Requisitos no funcionales</i>	21
2.4. FASE DE EXPLORACIÓN.....	22
2.4.1. <i>Historias de Usuario (HU)</i>	22
2.5. FASE DE PLANIFICACIÓN.....	27
2.5.1. <i>Estimación de esfuerzos por Historias de Usuario</i>	27
2.6. <i>Plan de Iteraciones</i>	27
2.6.1. <i>Plan de Entrega</i>	29
2.7. PATRÓN ARQUITECTÓNICO.....	29
2.8. PATRONES DE DISEÑO.....	31
2.9. TARJETAS CRC.....	33
2.9. DIAGRAMA DE CLASE.....	36
2.10. MODELO DE DATOS.....	37

CONSIDERACIONES PARCIALES	38
CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	39
3.1. FASE DE IMPLEMENTACIÓN	39
3.1.1. ITERACIÓN 1	39
3.1.2. ITERACIÓN 3	43
3.2. DIAGRAMA DE DESPLIEGUE.....	45
3.3. DIAGRAMA DE COMPONENTES	46
4.3. CASOS DE PRUEBAS DE ACEPTACIÓN	47
3.3. RESULTADOS	55
CONSIDERACIONES PARCIALES	55
CONCLUSIONES GENERALES	56
RECOMENDACIONES	57
REFERENCIAS.....	58
ANEXOS.....	61

ÍNDICE DE FIGURAS

FIGURA 1: PROCESO DE REGISTRO DE LA PROPIEDAD INTELECTUAL	7
FIGURA 2: MODELO-VISTA-CONTROLADOR PARA SYMFONY 2. (29).....	30
FIGURA 3: APLICACIÓN DE LA ARQUITECTURA EN LA CAPA MODELO.....	30
FIGURA 4: APLICACIÓN DE LA ARQUITECTURA EN LA CAPA CONTROLADOR.....	31
FIGURA 5: DIAGRAMA DE CLASES QUE REPRESENTA EL PATRÓN EXPERTO.....	31
FIGURA 6: DIAGRAMA DE CLASES QUE REPRESENTA EL PATRÓN CREADOR.....	32
FIGURA 7: DIAGRAMA DE CLASES QUE REPRESENTA EL PATRÓN CONTROLADOR.....	32
FIGURA 8: DIAGRAMA DE CLASES DEL SISTEMA.....	37
FIGURA 10: MODELO DE DATOS.....	38
FIGURA 11: DIAGRAMA DE DESPLIEGUE.....	45
FIGURA 11: DIAGRAMA DE COMPONENTES.....	46
FIGURA 12: PROTOTIPO DE LA PÁGINA PRINCIPAL DE LA APLICACIÓN.....	61
FIGURA 13: PROTOTIPO DE LA PÁGINA GESTIÓN DE ROLES DE LA APLICACIÓN.....	61
FIGURA 14: PROTOTIPO HU "AUTENTICAR USUARIO".....	62
FIGURA 15: PROTOTIPO HU "INSERTAR SOLICITUD".....	62
FIGURA 16: PROTOTIPO DE HU "LISTAR SOLICITUD".....	62
FIGURA 17: PROTOTIPO DE HU "REVISAR SOLICITUD".....	63
FIGURA 18: PROTOTIPO DE HU "ASIGNAR ROL".....	63
FIGURA 19: PROTOTIPO DE HU "CREAR ROL".....	63

ÍNDICE DE TABLAS

TABLA 1: SOFTWARE DE GESTIÓN DE LA PROPIEDAD INTELECTUAL.	8
TABLA 2: ANÁLISIS COMPARATIVO.....	¡ERROR! MARCADOR NO DEFINIDO.
TABLA 3: PERSONAL RELACIONADO CON EL SISTEMA.	19
TABLA 4: HU AUTENTICAR USUARIO.	22
TABLA 5: HU INSERTAR SOLICITUD.	22
TABLA 6: HU MODIFICAR SOLICITUD.....	23
TABLA 7: HU LISTAR SOLICITUD.	23
TABLA 8: HU REVISAR SOLICITUD.	24
TABLA 9: HU ENVIAR NOTIFICACIÓN.....	24
TABLA 10: HU ASIGNAR ROL.	24
TABLA 11: HU ELIMINAR SOLICITUD.	25
TABLA 12: HU ELIMINAR USUARIO.	25
TABLA 13: HU CREAR ROL.	25
TABLA 14: HU MODIFICAR ROL.	26
TABLA 15: HU ELIMINAR ROL.	26
TABLA 16: HU CERRAR SESIÓN.....	26
TABLA 17: ESTIMACIÓN DE ESFUERZOS POR HISTORIAS DE USUARIO.	27
TABLA 18: PLAN DE DURACIÓN DE LAS ITERACIONES.	28
TABLA 19: PLAN DE ENTREGAS.	29
TABLA 20: TARJETA CRC ROLESCONTROLLER.....	33
TABLA 21: TARJETA CRC SOLICITUDESCONTROLLER.....	33
TABLA 22: TARJETA CRC SEGURIDADCONTROLLER.....	34
TABLA 23: TARJETA CRC LDAP.	34
TABLA 24: TARJETA CRC ROL.....	35
TABLA 25: TARJETA CRC USER.	35
TABLA 26: TARJETA CRC AUTORES.	35
TABLA 27: TARJETA CRC SOLICITUDES.....	35
TABLA 28: TARJETA CRC ARCHIVOMU.	36
TABLA 29: TARJETA CRC ARCHIVOAV.....	36
TABLA 30: TAREAS POR HU DE LA ITERACIÓN 1.	39
TABLA 31: TAREA 1 DE LA HU 1.	40
TABLA 32: TAREA 2 DE LA HU 1.	40
TABLA 33: TAREA 3 DE LA HU 1.	40
TABLA 34: TAREA 1 DE LA HU 2.	41
TABLA 35: TAREA 2 DE LA HU 2.	41
TABLA 36: TAREA 3 DE LA HU 2.	41
TABLA 37: TAREA 1 DE LA HU 3.	42
TABLA 38: TAREA 2 DE LA HU 3.	42
TABLA 39: TAREA 3 DE LA HU 3.	42
TABLA 40: TAREA 3 DE LA HU 4.	42
TABLA 41: TAREA 4 DE LA HU 4.	43

TABLA 42: TAREAS POR HU DE LA ITERACIÓN 2.	43
TABLA 43: TAREA 1 DE LA HU 5.	43
TABLA 44: TAREA 2 DE LA HU 5.	44
TABLA 45: TAREA 3 DE LA HU 5	44
TABLA 46: TAREA 2 DE LA HU 6.	44
TABLA 47: TAREA 1 DE LA HU 7.	45
TABLA 48: PRUEBA DE ACEPTACIÓN PARA LA HU "AUTENTICAR USUARIO"	47
TABLA 49: PRUEBA DE ACEPTACIÓN PARA LA HU "INSERTAR SOLICITUD"	47
TABLA 50: PRUEBA DE ACEPTACIÓN PARA LA HU "MODIFICAR SOLICITUD"	48
TABLA 51: PRUEBA DE ACEPTACIÓN PARA LA HU "LISTAR SOLICITUD"	49
TABLA 52: PRUEBA DE ACEPTACIÓN PARA LA HU "ENVIAR NOTIFICACIÓN"	50
TABLA 53: PRUEBA DE ACEPTACIÓN PARA LA HU "REVISAR SOLICITUD"	50
TABLA 54: PRUEBA DE ACEPTACIÓN PARA LA HU " ASIGNAR ROL"	51
TABLA 55: PRUEBA DE ACEPTACIÓN PARA LA HU " ELIMINAR SOLICITUD"	52
TABLA 56: PRUEBA DE ACEPTACIÓN PARA LA HU " ELIMINAR SOLICITUD"	52
TABLA 57: PRUEBA DE ACEPTACIÓN PARA LA HU " CREAR ROL"	53
TABLA 58: PRUEBA DE ACEPTACIÓN PARA LA HU " MODIFICAR ROL"	53
TABLA 59: PRUEBA DE ACEPTACIÓN PARA LA HU " ELIMINAR ROL"	54

INTRODUCCIÓN

En las últimas décadas la invención de productos ha estado inmersa en un proceso continuo de internacionalización. Las empresas invierten grandes sumas de dinero en la creación de activos intangibles tales como: ideas, tecnologías, diseños, marcas, conocimientos en organización y modelos de negocios.

Los programas de computadora o software son ejemplos primordiales de dichas innovaciones. Estos juegan un papel fundamental en el proceso productivo del sector empresarial y en el desarrollo del mercado de las tecnologías de la información. Son numerosos los ejemplos que manifiestan su uso diario: programas de cálculo, gestión de bases de datos, gestión de proyectos, contabilidad, antivirus, entre otros.

En este contexto la Propiedad Intelectual (PI) juega un papel fundamental. El auge de nuevos países innovadores y la aspiración de proteger las inversiones en el extranjero ha traído consigo una creciente demanda de protección mediante las formas de PI.

Cuba no está exenta a esta situación; el avance vertiginoso en el desarrollo de software ha llevado a que se creen nuevos mecanismos de protección que aseguren el derecho de sus creadores.

La Red de Centros de la Universidad de las Ciencias Informáticas (UCI) es ejemplo claro del desarrollo de software en el país. El registro de la PI de sus productos es un proceso de gran importancia debido fundamentalmente a la protección legal que supone contar con la certificación de autoría de un resultado científico-productivo.

En el Centro de Informática Industrial de la mencionada Universidad, esta actividad se incluye dentro de las referidas a la Gestión de la Mercadotecnia. La misma involucra a todos los proyectos y miembros de proyectos de investigación y desarrollo con que cuenta el centro. Algunos inconvenientes hoy limitan la calidad y agilidad con que se desarrolla esta actividad.

Actualmente el CEDIN no cuenta con una aplicación que informatice todo el proceso de registro de la Propiedad Intelectual de los productos que en él se desarrollan.

El proceso de solicitud de registro de un producto se gestiona de forma personal. Dicho proceso implica el empleo del tiempo de la gestión y un intercambio de documentos por correo electrónico o físicamente, lo cual atenta contra la agilidad que debería primar. Se intercambia información con varios especialistas ajenos al Centro, dígase miembros del Grupo de Gestión Legal de la Dirección de Negocios. Se dictamina la información y la validan igualmente a través del correo electrónico, lo que en algunas ocasiones ha provocado la revisión y aceptación de información errónea o desactualizada.

La información de los registros es almacenada en una ubicación electrónica, en carpetas diferentes de acuerdo al estado en el que se encuentre la solicitud. Para ello, el especialista encargado define una organización que en algunos casos duplica la información, se pierde o se intercambia el estado en el que se encuentra la solicitud. Lo anterior redundaría en una

limitación importante en el seguimiento del estado de la solicitud, sin mencionar que no se documenta la fecha de realizada ni el momento en el que se encuentra lista para entregar.

Por otra parte, la forma en que se realiza el proceso actualmente no permite obtener estadísticas con facilidad. Entiéndase como estadísticas, la información referida a la cantidad de registros realizados, cantidad en proceso, plan de registros estimados, porcentos por departamentos u otras necesarias para evaluar el estado del registro de la propiedad intelectual. Actualmente no es posible obtenerla si no es a través del cálculo manual de estos importantes indicadores que introduce una incertidumbre en el dato a partir de un posible error humano. Todo lo anterior limita la toma de decisiones y la calidad y disponibilidad de la información que se presenta.

Por ello se plantea como **problema a resolver** de la presente investigación: ¿Cómo garantizar la gestión y control del registro de la propiedad intelectual de los productos del CEDIN?

Con el fin de dar solución al problema planteado se define como **objetivo general**: Desarrollar una aplicación informática para automatizar el proceso de registro de la propiedad intelectual del CEDIN.

En esta investigación el **objeto de estudio** lo constituye: La gestión de la Propiedad Intelectual.

Por todo lo anterior se precisa como **campo de acción**: Sistemas para la Gestión de Propiedad Intelectual de los productos informáticos.

Para darle cumplimiento al objetivo se trazaron las siguientes tareas:

- ❖ Definición de conceptos relacionados con la propiedad intelectual.
- ❖ Caracterización del proceso de registro de la propiedad intelectual en el mundo, Cuba y en el CEDIN.
- ❖ Valoración de las soluciones existentes que responden al problema de la investigación en alguna medida, sus limitaciones y fortalezas.
- ❖ Determinación de las principales herramientas, tecnologías, lenguajes y metodologías a utilizar para la construcción de la propuesta de solución.
- ❖ Definición del diseño de la solución propuesta.
- ❖ Implementación de la solución propuesta.
- ❖ Desarrollo del proceso de pruebas y validación del sistema implementado.

Fueron aplicados para la investigación los siguientes métodos de investigación científica:

De los **métodos teóricos**:

Método Analítico-Sintético: Se aplicó para el estudio de la Propiedad Intelectual para la fundamentación teórica de la solución a proponer.

Método Análisis Histórico-Lógico: Se utilizó en el análisis del basamento teórico, el estudio de las tecnologías a emplear y la descripción de herramientas informáticas de gestión de registros de la Propiedad Intelectual.

De los **métodos empíricos**:

Método Entrevista: Se utilizó en la realización de las consultas para el conocimiento del negocio a los individuos relacionados con el tema de los registros de Propiedad Intelectual.

Método Observación: Se utilizó en la investigación para determinar el problema a resolver, y comprender las necesidades del cliente y los procesos que se llevan a cabo actualmente para la automatización de los registros de la Propiedad Intelectual en el CEDIN.

El presente documento está estructurado en 3 capítulos y varios anexos, que incluye todo lo relacionado con el trabajo investigativo realizado, así como el análisis, diseño e implementación de la aplicación que se propone.

Capítulo 1 Fundamentación Teórica

Se ofrecen los elementos teóricos que sirven de base para la solución del problema. Se hace un estudio del estado del arte de las herramientas que sustentan la gestión de los registros de la propiedad intelectual. Además, se hace referencia a las diferentes tecnologías seleccionadas para la implementación del sistema.

Capítulo 2 Descripción y diseño de la solución propuesta.

Se identifican todos los requerimientos funcionales y no funcionales de la aplicación a implementar. Además, se muestran los principales procesos haciendo uso de las historias de usuario. Por otra parte se realiza la estimación para la implementación de la aplicación.

Capítulo 3 Implementación y validación de la solución propuesta de la aplicación

Se tratan los aspectos relacionados con la implementación de la solución propuesta. Además, se valida la solución propuesta mediante las pruebas unitarias y las pruebas de aceptación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se brinda una visión general de los aspectos relacionados con el proceso de propiedad intelectual, los principales conceptos relacionados con el problema, así como la descripción del objeto de estudio y campo de acción. También se muestran las principales tecnologías, metodologías, herramientas y lenguajes de programación utilizados para el desarrollo de la aplicación. Además de realizarse un análisis comparativo con otras herramientas existentes.

1.1. Propiedad Intelectual. Definiciones y categorías

Para entender que es la Propiedad Intelectual se debe ante todo, comprender el significado de las dos palabras que la componen. Si se hace referencia al Diccionario de la Real Academia Española “*Diccionario de la Lengua Española*” se define la palabra propiedad de la siguiente manera: “Derecho o facultad de gozar y disponer de una cosa con exclusión del ajeno arbitrio y de reclamar la devolución de ella si está en el poder del otro” (1).

El mismo diccionario define la palabra intelectual de la siguiente manera: “Perteneiente o relativo al entendimiento” (1).

Si se unen los dos significados anteriormente mencionados se llegaría a la conclusión que la Propiedad Intelectual se refiere a aquellas cosas creadas por el intelecto humano. A todas esas cosas que primeramente fueron ideas, luego un desarrollo del entendimiento humano y que concluye en la creación de un nuevo bien.

Jurídicamente la Propiedad Intelectual entra dentro de las ramas del Derecho. Esta rama regula las relaciones jurídicas de los hombres en el campo de los bienes intangibles. (2)

Las grandes economías desarrolladas buscan hoy en día una mayor protección de los valores intelectuales pues constituyen un importante eslabón en la obtención de grandes beneficios económicos y posterior elevación del nivel de vida de la sociedad.

Por otro lado, según la Declaración Mundial sobre la Propiedad Intelectual (votada por la Comisión Asesora de las políticas de la Organización Mundial de la Propiedad Intelectual (OMPI)), el 26 de junio del año 2000, se define como PI a: “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” (3).

La PI se divide en dos categorías fundamentales:

- ❖ La **Propiedad Industrial**, incluye las patentes de invenciones, las marcas, los diseños industriales y las indicaciones geográficas.
- ❖ El **Derecho de autor**, incluye obras literarias, tales como novelas, poemas y obras de teatro, películas, obras musicales, obras artísticas, tales como dibujos, pinturas,

fotografías y esculturas, diseños arquitectónicos, base de datos, multimedia y programas informáticos.

1.1.1. Propiedad Industrial

Según el Convenio de París, la propiedad industrial se refiere a: “los derechos que se adquieren por la creación de invenciones y diseños industriales, topografías de circuitos integrados, así como marcas, servicios de marcas, nombres y denominaciones comerciales, denominaciones de origen, indicaciones geográficas, secretos empresariales y los derechos relativos a la protección contra la competencia desleal, y debe ser aplicable no solo a la industria y el comercio propiamente dicho sino también a la agricultura, la industria extractiva y a todos los productos naturales y manufacturados”. (4).

Para el entendimiento de este tema se exponen una serie de conceptos vinculados a la Propiedad Industrial:

- ❖ **Patentes:** Se entiende por patente a “un derecho exclusivo concedido a una invención, que es el producto o proceso que ofrece una nueva manera de hacer algo, o una nueva solución técnica a un problema”. (5).

Las patentes proporcionan protección para la invención que se quiera hacer por parte de un titular. La protección se concede por un período determinado, variando este de las políticas que se tracen cada uno de los estados o países en el que se aplique dicho derecho. Además, la protección significa que la invención no podrá ser confeccionada, utilizada, distribuida o vendida sin una autorización previa del titular de la misma. Es el titular quien decide el uso de la misma dentro del período que se le fue dado. De ocurrir cualquiera de las prohibiciones anteriormente dichas, los tribunales tienen la potestad de sancionar las infracciones a la patente.

- ❖ **Marca:** Se entiende por marca a: “un signo distintivo que indica que ciertos bienes o servicios han sido producidos o proporcionados por una persona o empresa determinada”. (5)

Al igual que las patentes, las marcas proporcionan protección al titular de la misma, garantizándole el derecho exclusivo a utilizarla para identificar bienes o servicios, o autorizar su uso por otro individuo a cambio de un pago.

- ❖ **Diseño Industrial:** Se entiende por diseño industrial: “el aspecto ornamental o estético de un artículo. El diseño industrial puede consistir en rasgos en tres dimensiones, como la forma o la superficie de un artículo, o rasgos en dos dimensiones, como los diseños, las líneas o el color”. (5).

Los diseños industriales pueden ser aplicados tanto para instrumentos técnicos y médicos, relojes y joyas como para electrodomésticos, aparatos eléctricos, vehículos y estructuras arquitectónicas.

1.1.2.Derecho de Autor

Se define como derecho de autor a: “conjunto de normas jurídicas destinadas a regular y establecer los derechos y prerrogativas que tiene el autor sobre la obra producto de su intelecto, abarcando las facultades de autorizar la reproducción de este”. (6).

Las obras amparadas por el derecho de autor abarcan, aunque no exhaustivamente: novelas, poemas, obras de teatro, documentos de referencia, periódicos, programas informáticos, bases de datos, películas, composiciones musicales, coreografía, pinturas, dibujos, fotografías, escultura, arquitectura, publicidad, mapas y dibujos técnicos.

Entre las creaciones que la UCI registra se encuentran: producto de software o base de datos, marcas, diseños de comunicación visual, autorías DVD, obras fotográficas, literarias o científicas y audiovisuales.

¿Qué ventajas brinda aplicar el derecho de autor?

Las principales ventajas que presenta el uso de los derechos de autor se enmarcan en el hecho de proteger la propiedad intelectual de los seres humanos, comparándose este tipo de propiedad con la de carácter material en la cual existen documentos y leyes que protegen y legislan la propiedad privada.

Esta protección de tipo intelectual es de carácter personal y no solo se preocupa por la parte económica sino también por la moral del autor o titular de la obra. Existen leyes aprobadas en Cuba y el mundo que se enfocan netamente en los derechos de autor y son la parte legal para defender este tipo de derechos en la vía legal. El autor tiene libre albedrío para decidir si su obra es de carácter público o privado y también tiene protección de tipo legal por sus ideas. (2)

1.2. Registro de la Propiedad Intelectual en Cuba

En el mundo actual las empresas con mayor éxito, son las que han sabido implantar un sistema de gestión de la propiedad intelectual en su seno. Se trata no solo de aquellas que tengan registradas patentes, que hagan lo propio con sus marcas, o que protejan los diseños de mayor éxito, sino las que incluyan todo dentro de una estrategia. Esto abarca prácticamente a todas las áreas de una empresa y su implantación es absolutamente normal en las empresas más competitivas.

En Cuba el Sistema Nacional de Propiedad Industrial (SNPI) es la integración armónica de todo el régimen jurídico establecido en materia de Propiedad Industrial. Este régimen está integrado por la legislación de PI que establece la existencia de esta categoría de propiedad en forma de Derechos Exclusivos de carácter temporal, los actos de explotación económica y las modalidades de las creaciones del intelecto que son amparables por él. El sistema está dirigido por el Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMA) a través de la Oficina Cubana de la Propiedad Industrial (OCPI).

En materia de derecho de autor, Cuba cuenta con el Centro Nacional de Derechos de Autor, (CENDA). Este centro tiene como objetivo principal el de contribuir a que se creen las condiciones jurídicas y las premisas morales y materiales más propicias para el trabajo de escritores, artistas y científicos, así como para el fomento de sus obras. (7)

Por otro lado la Oficina Cubana de Propiedad Industrial (OCPI), como órgano rector del SNPI, fomenta la integración armónica de los organismos y empresas con las regulaciones nacionales e internacionales en esa materia y está encaminada a dotarlos de los conocimientos, pericias y destrezas necesarios para desempeñarse exitosamente en la gestión de la propiedad industrial, de acuerdo con sus respectivas responsabilidades.

En la UCI existe un procedimiento para registrar el patrimonio intelectual a través del ordenamiento y control de los requisitos necesarios para el registro de la Propiedad Intelectual. Las modalidades a proteger y por tanto sujetas a este procedimiento son: Productos de Software, Base de datos, obras Fotográficas, Literarias, Científicas, Audiovisuales, de Arte, Marcas, Diseños de Comunicación Visual y Autoría DVD, entre otras creaciones UCI.

En el siguiente diagrama se describe el procedimiento que es aplicado en la UCI para registrar la propiedad Intelectual de las creaciones que son implementadas:



Figura 1: Proceso de Registro de la Propiedad Intelectual.

1.3. Sistemas de gestión de la Propiedad Intelectual

En los últimos años el desarrollo de la informática ha alcanzado niveles excepcionales en la implementación de software para agilizar los procesos de gestión de cualquier índole. Los programas informáticos o software para la gestión de la Propiedad Intelectual son un ejemplo claro de ese desarrollo. La implementación de estos sistemas ha ayudado en el proceso de registro de la Propiedad Intelectual de las obras pertenecientes a un autor o titular determinado.

Un número creciente de departamentos legales de propiedad intelectual están utilizando soluciones de software de gestión para administrar sus activos de propiedad intelectual de forma eficaz. Estas soluciones deben tener presente una serie de características: debe proporcionar sólidas capacidades de flujo de trabajo y colaboración para automatizar todas

las operaciones manuales, reducir la cantidad de entrada de datos duplicados, y permitir la colaboración sin fisuras entre los inventores, los comités, departamentos de patentes de propiedad intelectual y bufetes de abogados.

Existen varios software de este tipo que han sido implementados a nivel mundial sobre los cuales se realizó un estudio que arrojó los siguientes resultados:

Tabla 1: Software de Gestión de la propiedad Intelectual.

Software	Fabricante	Características
Techframe's Darwin	TechFrame	El sistema está preparado para adaptarse a la especificidad del mercado de Propiedad Industrial a nivel mundial. Controla técnica y financieramente los derechos de Propiedad Industrial gestionados, y se da una importancia especial a la gestión sencilla de las múltiples variables relacionadas con todo el proceso de licenciamiento de Propiedad Industrial.
Apiges Pro	Grupo 10 Empresarial	Interrelación de todas las figuras asociadas a un proceso, tanto en los trámites oficiales con las Oficinas de Propiedad Industrial como en sus procedimientos jurídicos. Se puede consultar de forma automática y gratuita (en la mayoría de los casos) la situación de los expedientes y, en algunos asuntos, cumplimentar los formularios asociados a la solicitud u otros trámites, así como realizar pagos. Procesa cualquier modalidad de Propiedad Industrial, dominios, subvenciones y expedientes jurídicos

En Cuba también se utilizan aplicaciones para el proceso de gestión de Propiedad Intelectual. Los centros de investigación son los más marcados en este sentido. Tal es el caso del sistema implementado para el Centro Nacional de Sanidad Agropecuaria (CENSA) llamado SIPI (Sistema Interno de Propiedad Intelectual).

“Este sistema posibilita el reordenamiento y constancia de la actividad de Propiedad Intelectual, a través de una documentación generada y ordenada en cada proceso, que da respuesta a los requerimientos que se plantea la organización.

El sistema se inserta armónicamente en el régimen de calidad de la organización y se integra a la disciplina tecnológica de la misma garantizando una aplicación sostenida del régimen de Propiedad Intelectual que resulta trazable, define responsabilidades, permite su constante verificación y corrección, así como una revisión sistemática de la gestión en esta especialidad”. (8).

Como se pudo apreciar existen varias aplicaciones tanto a nivel mundial como nacional que gestionan el proceso de registro de la propiedad intelectual, pero son de difícil acceso por sus altos costos en el mercado internacional y son implementadas por compañías privadas. Por otra parte la herramienta implementada en el CENSA no satisface las necesidades del centro, debido a las diferencias de dominio de aplicación. Por un lado el CENSA certifica productos con el fin de preservar y elevar la sanidad animal, vegetal y humana, mientras que en el CEDIN certifica creaciones relacionadas con el mundo de la informática tales como, software, multimedia, base de datos, autorías DVD y otras.

1.4. Metodologías de desarrollo de software

El desarrollo de software no es una tarea fácil, por lo que ha surgido una alternativa que sirve de guía para llevar a cabo dicho proceso: **la metodología**. La misma surge ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto.

Hoy en día existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Ejemplo de ellas son las metodologías tradicionales, demostrando ser efectivas y necesarias en un gran número de proyectos y las ágiles centradas principalmente en el código fuente y menos orientada a la documentación.

1.4.1. Metodologías pesadas

“Son las más tradicionales; se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación, ya que pretende prever todo de antemano”. (9).

Entre las metodologías tradicionales o pesadas se destacan:

❖ **MSF (Microsoft Solution Framework):** “Es una flexible e interrelacionada serie de conceptos, modelos y prácticas de uso que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos”. (10).

Se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. “Entre sus **ventajas** se encuentra:

- ❖ Crea una disciplina de análisis de riesgos que ayuda y evoluciona con el proyecto.
- ❖ Vinculación con el cliente y orientado al trabajo en equipo.
- ❖ Tiene facilidad de soporte y mantenimiento.
- ❖ Permite la reutilización de componentes ya desarrollados en ciclos anteriores.
- ❖ Es un modelo enfocado a los requerimientos del usuario.
- ❖ Es una metodología que se puede ajustar a equipos de trabajo compuestas por 3 o más personas.

Desventajas:

- ❖ Al estar basado en tecnología Microsoft, trata de obligar a usar sus propias herramientas.
- ❖ Solicita demasiada documentación en sus fases.
- ❖ Si el análisis de riesgos se hace muy exhaustivo puede retardar el proyecto.
- ❖ Los precios de licencias, capacitación y soporte de Microsoft son caros.
- ❖ Alto grado de dependencias de tecnologías propietarias.” (11)

- ❖ **RUP (Rational Unified Process):** “Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.” (10)

Entre los **principales beneficios** que aporta el uso de esta metodología se encuentra:

- ❖ Permite desarrollar aplicaciones sacando el máximo provecho de las nuevas tecnologías, mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del software mediante una gestión sistemática de los riesgos.
- ❖ Proporciona guías explícitas para áreas tales como modelado de negocios, arquitectura Web, pruebas y calidad.
- ❖ Se integra estrechamente con herramientas Rational, permitiendo a los equipos de desarrollo aprovechar todas las ventajas de las características de los productos Rational, el Lenguaje de Modelado Unificado (UML) y otras prácticas óptimas de la industria.

1.4.2. Metodologías ágiles

“Las Metodologías Ágiles o “ligeras” constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de proyectos que las metodologías convencionales debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración”. (12).

Entre las metodologías ágiles se encuentran:

- ❖ **XP (Extreme Programming):** Es uno de los métodos más representativos dentro de las metodologías llamadas ligeras que lleva al extremo las prácticas para la consecución de sistemas funcionales y que cumplan las características del usuario cliente, surge como respuesta a la sobre planificación a la hora de generar software. Está basada en la simplicidad, la comunicación, la retroalimentación y la refactorización de código.

Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

La Programación Extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. XP considera que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso

deseable del desarrollo de proyectos. Ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

- ❖ **SCRUM:** “Es una forma de gestionar proyectos de software. No es una metodología de análisis, ni de diseño, como podría ser RUP, es una metodología de gestión del trabajo. Una de las características más importantes es que es muy fácil de explicar y de entender, lo que ayuda mucho a su implantación. Por otra parte SCRUM puede ser aplicado a distintos modelos de calidad (como podría ser CMMI) puesto que estos definen qué se tiene que hacer, es decir, definen que se tiene que gestionar el proyecto, pero no definen cómo. Ahí es donde entra SCRUM como modelo de gestión del proyecto”. (13).
- ❖ **SXP:** “Es un híbrido cubano de metodologías ágiles que tiene como base las metodologías SCRUM y XP que permiten actualizar los procesos de desarrollo de software para el mejoramiento de su producción. Consta de 4 fases: Planificación-Definición, Desarrollo, Entrega y Mantenimiento, cada una desglosada en flujos de trabajo y actividades que generan artefactos.

Esta metodología ayuda a fortalecer el trabajo en equipo, enfocados en una misma dirección, permitiendo además seguir de forma clara el avance de las tareas a realizar, a partir de la inserción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la producción, aumentando el nivel de interés del equipo”. (14)

1.4.3. Metodología de desarrollo de software utilizada

Para facilitar la elección de la metodología de desarrollo se listan un conjunto de características que servirán para comparar las ventajas y desventajas de RUP, XP, MSF.

Tabla 2: Análisis comparativo.

Característica	Metodología		
	RUP	XP	MSF
Desarrollo de aplicaciones web	Alto	Alto	Alto
Interacción del cliente con el equipo de desarrollo	Medio	Alto	Medio
Gestión de proyectos	Alto	Medio	Alto
Definición de requerimientos	Alto	Bajo	Alto
Facilidad para desarrollos de corta duración	Alto	Alto	Medio
Uso de herramientas libres	Alto	Alto	Bajo
Experiencia de equipo de desarrollo	Medio	Bajo	Medio

Después de haber analizado varias de las metodologías de desarrollo de software se decidió escoger la metodología XP, pues es la candidata ideal para guiar el proceso ingenieril de todo el ciclo del proyecto a desarrollar. Se adapta fácilmente ante el cambio, está orientada a las

personas y no a los procesos. El cliente forma parte del equipo de proyecto para establecer prioridades y resolver dudas en caso que existan. “Al realizarse ciclos muy cortos tras los cuales se muestran resultados, se minimiza el tener que rehacer partes que no cumplen con los requisitos y ayuda a los programadores a centrarse en lo que es más importante.” (15). Propone dar al cliente lo que desea cuando lo necesita y no en un tiempo lejano. Además, está enfocado a proyectos de poco tiempo de duración de desarrollo y a equipos pequeños.

1.5. Herramientas CASE

Las herramientas CASE (en inglés **Computer Aided Software Engineering**), traducido al español como Ingeniería de Software Asistida por Computadora, son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el diseño de proyectos, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. A continuación se ofrece una breve descripción de la herramienta CASE que se va a utilizar:

Visual Paradigm

“Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo software a través de la representación de todo tipo de diagramas. Constituye una herramienta de software libre de probada utilidad para el analista.

Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque orientado a objetos.

Se caracteriza por:

- ❖ Software libre.
- ❖ Disponibilidad en múltiples plataformas (Windows, Linux).
- ❖ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ❖ Capacidades de ingeniería directa e inversa.
- ❖ Disponibilidad de múltiples versiones, para cada necesidad.
- ❖ Licencia: gratuita y comercial.
- ❖ Fácil de instalar y actualizar.
- ❖ Soporte de UML versión 2.1.” (16)

Esta herramienta automatiza la documentación y ayuda a la reutilización del software, portabilidad y estandarización de la información. La misma es aplicada por política de la universidad, teniendo en cuenta el proceso de migración a software libre que permite garantizar la soberanía tecnológica.

1.6. Lenguaje de programación

1.6.1. Lenguajes de programación en el servidor

Los lenguajes de programación que se sitúan en el servidor son aquellos que son reconocidos, interpretados y procesados por la máquina que va a prestar un servicio concreto a un cliente. Los procesos que se llevan a cabo en el servidor, son realizados de manera transparente al cliente, lo cual garantiza que este no va a necesitar ningún software extra además del navegador, para poder satisfacer sus necesidades de uso con el sitio web.

Para la implementación de la aplicación se propone usar PHP como lenguaje de programación más adecuado para desarrollar el proyecto. Fue seleccionado ya que es un lenguaje gratuito que permite garantizar la soberanía tecnológica y de esa manera desarrollar productos que puedan ser licenciados bajo licencias de código abierto. Es independiente de plataforma, con una gran librería de funciones que cubren desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos. Por otra parte se vincula fácilmente con el marco de trabajo Symfony, pues está implementado bajo el lenguaje PHP 5.3 y es compatible con la base de datos MySQL. Es un lenguaje conocido mundialmente que ofrece disímiles ventajas entre las que se destacan:

- ❖ “Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- ❖ El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- ❖ Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.
- ❖ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL”. (17).

Por otra parte se vincula fácilmente con el marco de trabajo Symfony, pues está implementado bajo el lenguaje PHP 5.3.

1.6.2. Lenguajes de programación para servirle al cliente

Un lenguaje de programación del lado del cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio. Dentro de estos se encuentran HTML (Hyper Text Markup Language), JavaScript, CSS (Cascading Style Sheets) y otros.

A continuación se describe los lenguajes que fueron empleados para la implementación de la solución informática.

1.6.2.1. Lenguaje de Marcado de Hipertexto (HTML)

El HTML (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web). Este lenguaje no es propiamente un lenguaje de programación como C++, Visual Basic y otros, sino un sistema de etiquetas.

Entre las principales ventajas que aporta el uso de este lenguaje de programación se encuentra:

- ❖ “Sencillo que permite describir hipertexto.
- ❖ Texto presentado de forma estructurada y agradable.
- ❖ No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web.
- ❖ Archivos pequeños.
- ❖ Despliegue rápido.
- ❖ Lenguaje de fácil aprendizaje.
- ❖ Lo admiten todos los exploradores” (18).

1.6.2.2. JavaScript

JavaScript es un lenguaje de programación que realiza acciones dentro del ámbito de una página web. Su compatibilidad con la mayoría de los navegadores modernos, lo posiciona como el lenguaje de programación del lado del cliente más utilizado.

“JavaScript soporta el Modelo de Objetos de Documento (DOM, Document Object Model) que es el conjunto de objetos predefinidos que permite acceder a todos los elementos de una página y a ciertas características específicas del navegador”. (19) Además posibilita la creación de efectos llamativos en las páginas web e interacción con el usuario.

El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estas acciones, de modo que el mayor recurso con que cuenta este lenguaje es el navegador.

“Desde su aparición, JavaScript siempre fue utilizado de forma masiva por la mayoría de los sitios de Internet. La aparición de Flash disminuyó su popularidad, sin embargo, la aparición de las aplicaciones Ajax programadas con JavaScript le ha devuelto una popularidad sin igual dentro de los lenguajes de programación web”. (20)

El Ajax sumado a JavaScript permite ejecutar instrucciones como respuestas a las acciones del usuario (eventos), son dos habilidades muy explotadas en el desarrollo web actualmente con la inclusión en su código fuente de la librería o framework jQuery.

1.6.2.3. Hojas de estilo en cascada (CSS)

Las hojas de estilo en cascada (en inglés *Cascading Style Sheets* o CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML y por extensión en XHTML.

Las hojas de estilo representan un avance importante para los diseñadores de páginas web, al darles un mayor rango de posibilidades para mejorar la apariencia de sus páginas. En los entornos científicos en que la web fue concebida, la gente estaba más preocupada por el contenido de sus páginas que por su presentación.

1.7. Sistemas de Gestión de Base de Datos

En los últimos años, el software de bases de datos ha experimentado un auge extraordinario, a raíz de la progresiva informatización de casi la totalidad de las empresas de hoy en día. No es extraño pues, que existan multitud de sistemas de gestión de bases de datos (SGBD), programas que permiten manejar la información de modo más sencilla. Algunos de ellos son Oracle, Microsoft SQL Server, entre otros.

En el mundo del software libre, siempre que se necesita algo, tarde o temprano se implementa. Un ejemplo claro es MySQL, gestor muy usado en la web o PostgreSQL.

1.7.1.PostgreSQL

“PostgreSQL es uno de los mejores Sistemas Gestores de Base de Datos utilizado principalmente por la comunidad de software libre por sus cualidades: soporta el lenguaje común de acceso a los datos: SQL. Es multiplataforma y posee buenas interfaces de instalación y administración. Aproxima los datos a un modelo Objeto Relacional, y es capaz de manejar completas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multiversión, soporte multiusuario, transacciones y optimización de consultas”. (21)

Entre las ventajas que ofrece:

- ❖ Restauración de bases de datos en procesos paralelos, que acelera recuperación de un respaldo hasta 8 veces.
- ❖ Privilegios por columna, que permiten un control más granular de datos confidenciales.
- ❖ Nuevas herramientas de monitoreo de consultas que le otorgan a los administradores mayor información sobre la actividad del sistema.
- ❖ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- ❖ Soporta el uso de índices, reglas y vistas.

1.7.2.MySQL

“MySQL es un sistema de gestión de bases de datos relacional y multiusuario. Utilizado principalmente en aplicaciones web, como Drupal o phpBB. Es capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización. MySQL es un software de código abierto, licenciado bajo la Licencia Pública General (GPL). Es nombrada uno de los 10 mejores productos de código abierto de 2011 por CRN (del inglés **Computer Reseller News**)”. (22)

Entre sus ventajas se encuentra:

- ❖ Tablas hash en memoria temporales.
- ❖ Completo soporte para operadores y funciones en cláusulas select y where.
- ❖ Completo soporte para cláusulas group by y order by, soporte de funciones de agrupación.
- ❖ Ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- ❖ Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.
- ❖ Las principales propiedades de la web se amplían perfectamente con MySQL.

1.7.3.Sistema de Gestión de Bases de Datos a utilizar

Luego de realizar un estudio de los diferentes SGBD, se manifiesta que MySQL es el indicado para el trabajo con la BD del sistema a implementar. Esta elección se basa en que SQL Server no es multiplataforma; para su uso es necesario pagar importes por su licencia, lo cual no cumple con las políticas de la Universidad de migrar al software libre.

El uso de MySQL en herramientas web es cada vez mayor debido a su simplicidad. El hecho de ser un producto de código abierto, sin costos de licencia, convierte a MySQL en una alternativa extremadamente adecuada para las empresas que buscan un ahorro significativo.

1.8. Entorno de desarrollo integrado

Un Entorno de Desarrollo Integrado (en inglés *Integrated Development Environment* o IDE) es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código.” (23).

Son muchos los entornos que son utilizados en el mundo del desarrollo web. En lo adelante se caracteriza el entorno de desarrollo a ser aplicado.

“**NetBeans** es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. La plataforma NetBeans es un IDE de código abierto escrito completamente en Java. Soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web y aplicaciones móviles). También tiene soporte para varios lenguajes como PHP, JavaScript, HTML, CSS. Es un producto libre y gratuito sin restricciones de uso. Esta plataforma permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Además, es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación”. (24).

Es un entorno integrado para el desarrollo de aplicaciones web. La plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones. Proporciona extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

1.9. Framework

“En el desarrollo de software, un marco de trabajo o framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.” (25)

“Symfony es un framework de PHP basado en la arquitectura MVC (Modelo-Vista-Controlador). Su primera versión surgió por el año 2005 como un proyecto de software libre el cual se ha convertido en uno de los framework más populares de PHP que existe en la actualidad.” (26)

“Symfony fue diseñado para optimizar el desarrollo de aplicaciones web, proporcionando herramientas para agilizar aplicaciones complejas y guiando al desarrollador a acostumbrarse al orden y buenas prácticas dentro del proyecto. El mismo puede ser utilizado para otros tipos de desarrollos no orientados a la Web.” (27)

Este framework reutiliza conceptos y desarrollos exitosos de terceros y los integra como librerías. Entre ellos encontramos que integra plenamente uno de los framework ORM más importantes dentro de los existentes para PHP llamado Doctrine. El mismo es el encargado de la comunicación con la base de datos, permitiendo un control casi total de los datos sin importar si estamos hablando de MySQL, PostgreSQL, SQL server, Oracle, entre otros motores.

1.10. Herramientas, metodologías y lenguajes a utilizar

Luego del estudio realizado se determinó que el sistema estará guiado por la metodología de desarrollo XP. Para modelar el proceso se utilizará como herramienta CASE Visual Paradigm UML 8.0. Se hará uso del lenguaje de programación PHP. Además, se empleará el marco de trabajo Symfony aprovechando las facilidades que brinda en el desarrollo web. Como Sistema de Gestión de Bases de Datos se aplica MySQL teniendo en cuenta las políticas de migración de la universidad a software libre además de la integración con PHP.

Consideraciones Parciales

1. El estudio de los conceptos relacionados con la propiedad intelectual y del proceso de registro de la misma en la Universidad de las Ciencias Informáticas permitió crear la base para determinar las características que debe tener el sistema a desarrollarse.
2. En el mercado internacional se encuentran aplicaciones encargadas de la gestión de la Propiedad Intelectual, pero son desarrolladas por empresas privadas y su precio en

los mercados son demasiado elevados, por lo que no resulta factible aplicarlo en el Centro de Informática Industrial.

3. El estudio sobre las principales herramientas, tecnologías y metodologías permitió seleccionar las más adecuadas para ser usadas en el desarrollo de la solución informática.

CAPÍTULO 2: DESCRIPCIÓN Y DISEÑO DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se procede a realizar la descripción de la propuesta de solución. Se enumeran los requisitos funcionales y no funcionales del sistema. Además se explica el uso de Patrones de diseño para la implementación y el patrón arquitectónico. Se describen las fases de exploración y planificación de la metodología seleccionada. Son definidas las tarjetas CRC para la posterior implementación. Por último se diseñará el modelo de datos de la base de datos, el diagrama de clases y el diagrama de componentes de la aplicación.

2.1. Descripción de la propuesta de solución

La propuesta de solución consiste en una aplicación web para gestionar la propiedad intelectual de los productos que se desarrollan en el CEDIN. A través de la misma el autor asignado podrá hacer el registro y subir la documentación correspondiente a la solicitud. Además, se puede llevar el seguimiento de todo el proceso, dígase revisión a las planillas y documentos por parte de los factores involucrados y correcciones a las mismas siempre manteniendo la última versión de la documentación en el sitio.

El autor por su parte estará informado del estado por la cual su solicitud está pasando, mediante un sistema de notificación vía correo. Además, podrá modificar su solicitud y tendrá la posibilidad de asignar un nuevo usuario a la misma.

Por otra parte la aplicación cuenta con un sistema de autenticación, mediante roles que fueron predefinidos según los actores involucrados en el proceso. De esta manera se diferencia el entorno de trabajo con el que trabajará cada rol.

2.2. Personal relacionado con el sistema

La aplicación va dirigida al Grupo de Inteligencia Empresarial del CEDIN para ser aplicado en la certificación de los productos que en él se desarrollan. En la siguiente tabla se describen los actores que interactúan con la aplicación web:

Tabla 3: Personal relacionado con el sistema.

Personal relacionado con el sistema	
Actores	Justificación
Administrador	Es el encargado (a) de llevar el control total de la aplicación web.
Autor de la solicitud	Es el encargado (a) de realizar el registro de solicitud de certificación del producto.
Asesor Comercial	Es el encargado (a) de la revisión de la solicitud y documentación.

Especialista de Servicios Legales	Personal de la Dirección de Servicios Legales (DSL) interesada en conocer sobre las solicitudes registradas.
Invitado	Personal interesado en conocer sobre las solicitudes registradas en el centro.

2.3. Requisitos del sistema

“El principal problema al que nos enfrentamos en el desarrollo de sistemas es la ingeniería de requerimientos. Esta trata de establecer lo que el sistema debe hacer, sus propiedades emergentes deseadas y esenciales, y las restricciones en el funcionamiento del sistema. Por lo tanto puede considerarse la ingeniería de requerimientos como el proceso de comunicación entre los clientes y los desarrolladores del mismo.

Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar información”. (28).

2.3.1. 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”. (28). A continuación se muestran los requisitos funcionales del sistema:

RF 1: Autenticar usuario.

RF 2: Insertar solicitud.

RF 3: Modificar la solicitud.

RF 4: Listar solicitudes.

RF 5: Revisar solicitud.

RF 6: Enviar notificación.

RF 7: Asignar rol.

RF 8: Eliminar solicitud.

RF 9: Eliminar usuario.

RF 10: Crear rol.

RF 11: Modificar rol.

RF 12: Eliminar rol.

RF 13: Cerrar sesión.

2.3.2.Requisitos no funcionales

“Los requerimientos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares”. (28). Para la aplicación fueron capturados los siguientes requisitos no funcionales:

Usabilidad:

- ❖ **Facilidad de uso por parte de los usuarios:** La aplicación debe presentar una interfaz que permita fácil interacción entre las funcionalidades, de manera que el usuario pueda llegar de manera rápida y efectiva a la información buscada.
- ❖ **Emplear perfiles de usuario:** Diferenciar las interfaces y opciones para los usuarios que accedan al módulo según los diferentes roles que estos tengan dentro del mismo.
- ❖ La aplicación tendrá siempre visible la opción de Ayuda, lo que posibilitará una mejor explotación de sus funcionalidades por parte de los usuarios.

Rendimiento:

- ❖ Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 3 segundos para las actualizaciones y 10 para las recuperaciones.

Software:

- ❖ **Para el cliente:**
 - Sistema operativo XP o superior, o Linux.
 - Utilizar navegador (Mozilla Firefox o Google Chrome).
- ❖ **Para el servidor:**
 - Sistema operativo Windows o Linux en cualquiera de sus distribuciones.
 - Servidor de base de datos MySQL.
 - Servidor con módulo PHP disponible (WAMP, XAMPP, u otros).

Hardware:

- ❖ Capacidad de disco duro superior a 10 Giga bytes. Se requiere un mínimo de 512 MB de RAM y 1.6 GHz de velocidad de procesamiento.

Seguridad:

- ❖ Los usuarios deben autenticarse antes de interactuar con el sistema.
- ❖ El sistema deberá garantizar el acceso controlado a la información. Este debe influir sobre cómo se presentan las interfaces para cada usuario dependiendo del nivel de acceso a la información.

2.4. Fase de Exploración

La exploración es la etapa del proceso de desarrollo de software que propone la metodología XP para comenzar la construcción de un producto. En ella el usuario asegura que exista suficiente material en las historias de usuario (HU) para hacer una primera liberación del producto. El equipo de desarrollo se relaciona con cada una de las tecnologías y herramientas que utilizará en el desarrollo del sistema y también explora la posibilidad de conformar la arquitectura. Un aspecto clave en el esclarecimiento de las dudas sobre los procesos a automatizar es que, desde el mismo inicio del proyecto, un miembro del equipo del trabajo del cliente, se adiciona al de desarrolladores. (15)

2.4.1. Historias de Usuario (HU)

“Una historia de usuario es una funcionalidad o parte de una funcionalidad en la metodología XP. Debe ser entendible por los clientes y los desarrolladores, posible de probar, de valor para el cliente y lo suficientemente pequeña como para que los desarrolladores puedan implementar media docena en una iteración”. (15).

Tabla 4: HU Autenticar usuario.

Historias de Usuario	
Número: 1	Usuario: Todos
Nombre Historia: Autenticar usuario.	
Prioridad en el negocio: Alta	Nivel de complejidad: Alta
Tiempo de estimación: 0.3	Iteración asignada: 1
Descripción: El sistema debe permitir la autenticación de usuario para acceder al entorno de trabajo del mismo mediante usuario y contraseña. Se debe poner en práctica el Protocolo Ligero de Acceso a Directorios (LDAP) de la universidad, para la verificación de usuarios en el dominio. Automáticamente al usuario autenticado se le asigna el rol de invitado.	
Prototipo: Ver anexo Figura 12.	

Tabla 5: HU Insertar solicitud.

Historias de Usuario	
Número: 2	Usuario: Autor.
Nombre Historia: Insertar solicitud.	
Prioridad en el negocio: Alta	Nivel de complejidad: Alta
Tiempo de estimación: 1	Iteración asignada: 1

Descripción: El sistema debe permitir la inserción de solicitudes por parte de un usuario. Se debe generar un formulario en la que el autor deberá llenar los datos requeridos. El mismo debe recoger la siguiente información: Nombre de proyecto, Nombre del producto, versión, facultad, centro, cliente, organismo, clasificación, tipo de registro, fecha, nombre y apellidos del solicitante, usuario y cargo, un resumen del producto. Se deben obtener los datos de los autores que tienen derecho sobre la aplicación a registrar (Nombre, Apellidos, Rol, Entidad, Carnet de Identidad). Además debe permitir la subida de documentos (Manual de Usuario y Acta de Validación), que permitirán completar el proceso de creación de la solicitud. Todos los datos insertados deben ser validados acorde al tipo de dato que soporten. Al finalizar la acción se le asigna automáticamente el estado **Iniciado**. Todos los campos deben soportar texto, exceptuando el campo **Versión** y el **Carnet de Identidad** que son numéricos. El **Carnet de Identidad** debe tener 11 dígitos. Al concluir la inserción de la solicitud se debe mostrar un mensaje: **“Su solicitud ha sido creada satisfactoriamente”**.

Prototipo: Ver anexo Figura 13.

Tabla 6: HU Modificar solicitud.

Historias de Usuario	
Número: 3	Usuario: Autor.
Nombre Historia: Modificar solicitud.	
Prioridad en el negocio: Alta	Nivel de complejidad: Media
Tiempo de estimación: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir la modificación de la solicitud en caso de haber existido algún error por parte de quién realiza la acción. Para ello debe generarse el formulario de la solicitud a modificar con los datos que fueron insertados inicialmente. Los datos a modificar deben ser actualizados en la base de datos del sistema. Se debe mostrar un mensaje: “Su solicitud ha sido modificada satisfactoriamente” .	

Tabla 7: HU Listar solicitud.

Historias de Usuario	
Número: 4	Usuario: Todos.
Nombre Historia: Listar solicitud.	
Prioridad en el negocio: baja	Nivel de complejidad: Media
Tiempo de estimación: 0.5	Iteración asignada: 3

Descripción: El sistema debe listar las solicitudes creadas por el usuario que realiza la petición para el rol de Autor. En el caso de los roles Especialista, Invitado, Administrador y Asesor se muestran todas las solicitudes realizadas en el centro. Este listado se muestra ordenado teniendo en cuenta la fecha de creación de forma descendente.

Prototipo: Ver anexo Figura 14.

Tabla 8: HU Revisar solicitud.

Historias de Usuario	
Número: 5	Usuario: Asesor.
Nombre Historia: Revisar solicitud.	
Prioridad en el negocio: Alta	Nivel de complejidad: Alta
Tiempo de estimación: 2	Iteración asignada: 1
Descripción: El sistema debe listar las solicitudes. Para ejecutar la revisión, se debe primero modificar el estado de la misma, quedando En revisión . Luego se descarga la documentación que será revisada por el Asesor Comercial (Manual de Usuario y Acta de Validación y se genera el documento pdf que contiene los datos de la solicitud). Una vez arreglados los errores que existan se podrán subir los archivos corregidos. Al finalizar esta acción el Asesor asigna el estado Aceptada a la solicitud.	
Prototipo: Ver anexo Figura 15.	

Tabla 9: HU Enviar notificación.

Historias de Usuario	
Número: 6	Usuario: Aplicación.
Nombre Historia: Enviar notificación.	
Prioridad en el negocio: Baja	Nivel de complejidad: Medio
Tiempo de estimación: 0.5	Iteración asignada: 3
Descripción: El sistema debe notificar a los usuarios de manera automática el estado en que se encuentran las solicitudes que fueron generadas con anterioridad. Dichas notificaciones se realizarán mediante correo electrónico.	

Tabla 10: HU Asignar rol.

Historias de Usuario	
Número: 7	Usuario: Administrador
Nombre Historia: Asignar rol.	
Prioridad en el negocio: Bajo	Nivel de complejidad: Media
Tiempo de estimación: 0.5	Iteración asignada: 3

Descripción: El sistema debe mostrar un listado con los usuarios que están registrados, así como el rol que tiene asignado cada uno. A partir de esto el administrador será capaz de cambiar el rol haciendo uso de una lista desplegable con los roles predefinidos (Administrador, Autor, Asesor, Especialista, Invitado).

Prototipo: Ver anexo Figura 16.

Tabla 11: HU Eliminar solicitud.

Historias de Usuario	
Número: 8	Usuario: Autor
Nombre Historia: Eliminar solicitud.	
Prioridad en el negocio: Baja	Nivel de complejidad: Media
Tiempo de estimación: 0.5	Iteración asignada: 3
Descripción: El sistema debe mostrar un listado con las solicitudes registradas por el usuario que hace la petición, dándole la posibilidad de eliminarlas. Al realizar la petición de eliminación, el sistema debe mostrar el mensaje de confirmación: “ ¿Está seguro que desea eliminar la solicitud de forma permanente? ” para que el usuario confirme la ejecución de la acción.	

Tabla 12: HU Eliminar usuario.

Historias de Usuario	
Número: 9	Usuario: Administrador
Nombre Historia: Eliminar usuario.	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe mostrar un listado con los usuarios que están registrados en la aplicación, dándole la posibilidad al administrador de eliminarlos. Al realizar la petición de eliminación, el sistema debe mostrar el mensaje de confirmación: “ ¿Está seguro que desea eliminar el usuario de forma permanente? ” para que el administrador confirme la ejecución de la acción.	

Tabla 13: HU Crear rol.

Historias de Usuario	
Número: 10	Usuario: Administrador
Nombre Historia: Crear rol.	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de estimación: 0.5	Iteración asignada: 2

Descripción: El sistema debe permitir la creación de roles por parte del administrador. Se debe generar un formulario en el que el administrador deberá llenar los datos requeridos. El mismo debe recoger la siguiente información: Nombre del rol. El dato insertado debe ser validado acorde al tipo de dato que soporte y verificar si existe algún rol creado con el nombre del nuevo rol. El campo debe soportar texto.

Prototipo: Ver anexo Figura 17.

Tabla 14: HU Modificar rol.

Historias de Usuario	
Número: 11	Usuario: Administrador
Nombre Historia: Modificar rol.	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe permitir la modificación del rol. Para ello debe generarse el formulario del rol a modificar con el dato que fue insertado inicialmente. El dato a modificar debe ser actualizado en la base de datos del sistema.	

Tabla 15: HU Eliminar rol.

Historias de Usuario	
Número: 12	Usuario: Administrador
Nombre Historia: Eliminar rol.	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de estimación: 0.5	Iteración asignada: 2
Descripción: El sistema debe mostrar un listado con los roles que están registrados en la aplicación, dándole la posibilidad al administrador de eliminarlos. Al realizar la petición de eliminación, el sistema debe mostrar el mensaje de confirmación: “ ¿Está seguro que desea eliminar el usuario de forma permanente? ” para que el administrador confirme la ejecución de la misma. Una vez ejecutada el sistema debe verificar si existen usuarios con ese rol asignado. De suceder lo anterior se muestra un formulario para que se les asigne otro rol a dichos usuarios. El formulario debe ser validado.	

Tabla 16: HU Cerrar sesión.

Historias de Usuario	
Número: 13	Usuario: Todos
Nombre Historia: Cerrar sesión.	
Prioridad en el negocio: Alta	Nivel de complejidad: Alta

Tiempo de estimación: 0.2	Iteración asignada: 1
Descripción: El sistema debe permitir al usuario cerrar la sesión en la que se encuentra y ser redireccionado a la página de inicio, permitiendo que otros usuarios puedan autenticarse.	

2.5. Fase de planificación

El propósito de esta fase es llegar a un acuerdo entre el cliente y el desarrollador sobre la posible fecha de entrega de cada grupo de funcionalidades. Se hace una estimación del esfuerzo que requerirá por parte de los desarrolladores de dar solución a cada una de las HU.

Se toma como unidad de medida el punto, el cual se considera como una semana (5 días) de trabajo, en la cual el desarrollador trabajará de manera ininterrumpida para cumplir con la planificación trazada. (15)

2.5.1. Estimación de esfuerzos por Historias de Usuario

Tabla 17: Estimación de esfuerzos por Historias de Usuario.

Historias de Usuario	Estimación por puntos
Autenticar usuario	0.3
Insertar solicitud	1
Modificar solicitud	0.5
Listar solicitud	0.5
Revisar solicitud	2
Enviar notificación	0.5
Asignar rol	0.5
Eliminar solicitud	0.5
Eliminar usuario	0.5
Crear rol	0.5
Modificar rol	0.5
Eliminar rol	0.5
Cerrar sesión	0.2

2.6. Plan de Iteraciones

Después de identificado cada una de las HU y haber analizado la estimación de esfuerzos, se procede entonces a confeccionar el plan de iteraciones. El mismo tiene como objetivo definir las HU que serán implementadas en cada una de las iteraciones. A continuación se muestra como fueron desplegadas:

- ❖ **Iteración 1:** Se implementa las HU de mayor importancia para el funcionamiento de la aplicación, de las cuales se encuentran la 1, 2, 3 y 5, 13. Al terminar la iteración el usuario podrá autenticarse a la aplicación, cerrar sesión, insertar, modificar y revisar solicitudes. De esta manera se obtendrá la primera versión funcional de la aplicación. De la misma pueden derivarse nuevos requerimientos funcionales. Por último se realizan las pruebas pertinentes a las funcionalidades que fueron implementadas.
- ❖ **Iteración 2:** Se implementa las HU de prioridad media para la aplicación, en las cuales se incluyen la 9, 10, 11 y 12. Al finalizar la iteración se podrá eliminar usuarios registrados en la aplicación y crear, modificar y eliminar roles. Luego se pasará a solucionar los errores y no conformidades que fueron detectadas en la iteración anterior. Finalmente se ponen a prueba las nuevas funcionalidades implementadas.
- ❖ **Iteración 3:** Se implementa las HU restantes, entre las que se incluyen la 4, 6, 7 y 8. El usuario será capaz de enviar notificaciones al usuario que hace la inserción para que conozca el estado por las que va pasando las solicitudes. Además podrá asignar roles a los usuarios registrados y listar y eliminar solicitudes creadas. Al finalizar esta iteración, se constará con una versión estable del producto con todas las funcionalidades implementadas y probadas.

Tabla 18: Plan de duración de las iteraciones.

Iteración	Historias de Usuario	Estimación por puntos
1	Autenticar usuario	4
	Insertar solicitud	
	Modificar solicitud	
	Revisar solicitud	
	Cerrar sesión	
2	Eliminar usuario	2
	Crear rol	
	Modificar rol	
	Eliminar rol	
3	Listar solicitud	2
	Enviar notificación	
	Asignar rol	
	Eliminar solicitud	
Total		8

2.6.1. Plan de Entrega

A partir de las iteraciones realizadas anteriormente se procede a elaborar el plan de entregas. El mismo tiene como objetivo definir las diferentes entregas y orden que tendrán las mismas.

Tabla 19: Plan de entregas.

Historias de Usuario	Final 1era Iteración (28 de febrero)	Final 2da Iteración (31 de marzo)	Final 3era Iteración (9 de abril)
Autenticar usuario	V1.0		
Insertar solicitud	V1.0		
Modificar solicitud	V1.0		
Revisar solicitud	V1.0		
Cerrar sesión	V1.0		
Eliminar usuario		V1.1	
Crear rol		V1.1	
Modificar rol		V1.1	
Eliminar rol		V1.1	
Listar solicitud			V1.2
Enviar notificación			V1.2
Asignar rol			V1.2
Eliminar solicitud			V1.2

2.7. Patrón Arquitectónico

La arquitectura de software es la estructura de un sistema, que contiene elementos del software, las propiedades externas visibles de esos elementos, y las relaciones entre ellos. Una correcta definición de la arquitectura permite facilitar el diseño y construcción de los componentes y sus relaciones.

Entre los patrones arquitectónicos empleados en la construcción de aplicaciones web se encuentra el Modelo-Vista-Controlador (MVC), que separa el modelo de datos, la lógica de control y las interfaces de usuario.

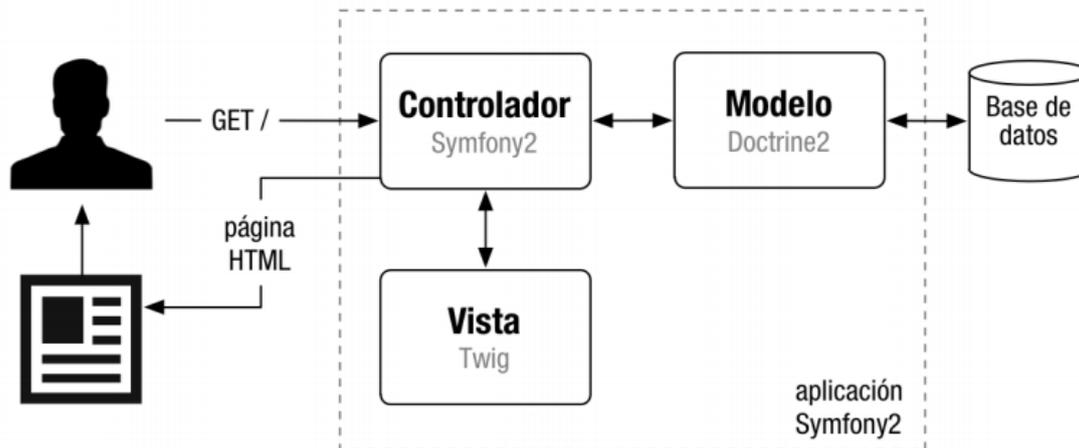


Figura 2: Modelo-Vista-Controlador para Symfony 2. (29)

El **modelo** define la lógica de negocio, incluyendo la base de datos. Symfony2 guarda todas las clases y archivos relacionados con el modelo en el directorio 'lib/model/'.

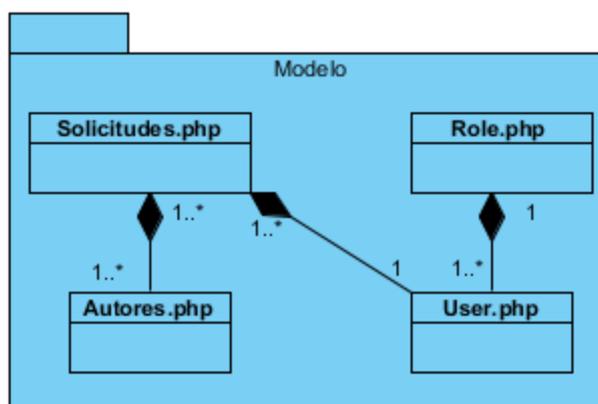


Figura 3: Aplicación de la arquitectura en la capa Modelo.

La **vista** es lo que utilizan los usuarios para interactuar con la aplicación (los gestores de plantillas pertenecen a esta capa). En Symfony2 la capa de la vista está formada principalmente por plantillas en PHP. Estas plantillas se guardan en varios directorios llamados 'templates/'.

La vista recibe los datos de la controladora y luego los organiza para ser presentada al usuario de forma organizada en el navegador web. Las etiquetas que son usadas en Symfony para el diseño de las plantillas permiten que sea flexible para los diseñadores, incluso tiene estructuras de datos como if, por si es necesaria una presentación lógica de los datos.

El **controlador** es un bloque de código que realiza llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario. Todas las peticiones se canalizan a través de los controladores frontales, que delegan todo el trabajo en las acciones, las que se agrupan en módulos.

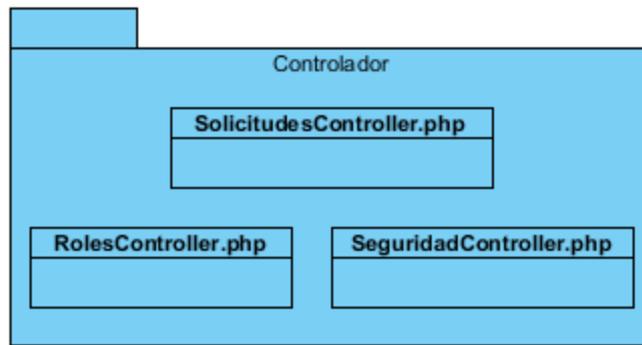


Figura 4: Aplicación de la arquitectura en la capa Controlador.

2.8. Patrones de diseño

Un patrón de diseño nombra, abstrae e identifica los aspectos clave de un diseño estructurado común, que lo hace útil para la creación de diseños orientados a objetos reutilizables. Definen una posible solución correcta para un problema de diseño dentro de un contexto dado, describiendo las cualidades invariantes de todas las soluciones. (30)

En el diseño del software orientado a objeto el uso de los patrones GRASP acrónimo del inglés **General Responsibility Assignment Software Patterns** (Patrones Generales de Software para asignar responsabilidades) tiene una gran importancia, debido a que dan solución a muchos de los problemas que se pueden presentar a la hora de programar.

Para el diseño de la aplicación web se tuvieron en cuenta los siguientes patrones GRASP:

El **patrón Experto** es utilizado para mapear la base de datos de la aplicación. Dicho patrón fue aplicado para generar las clases que controlarán toda la información que es almacenada en la base de datos. Las mismas poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y a su vez contienen la información necesaria de la tabla a la que está asociada.

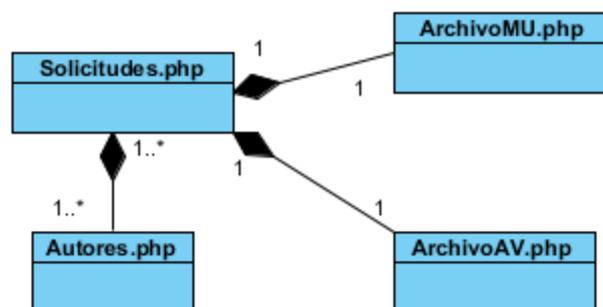


Figura 5: Diagrama de clases que representa el Patrón Experto.

Otro de los patrones aplicados es el **Creador**. El mismo ayuda a identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases. Tiene la información

necesaria para realizar la creación del objeto que es una de las actividades más comunes en un sistema orientado a objetos. Almacena o maneja varias instancias de la clase. (31)

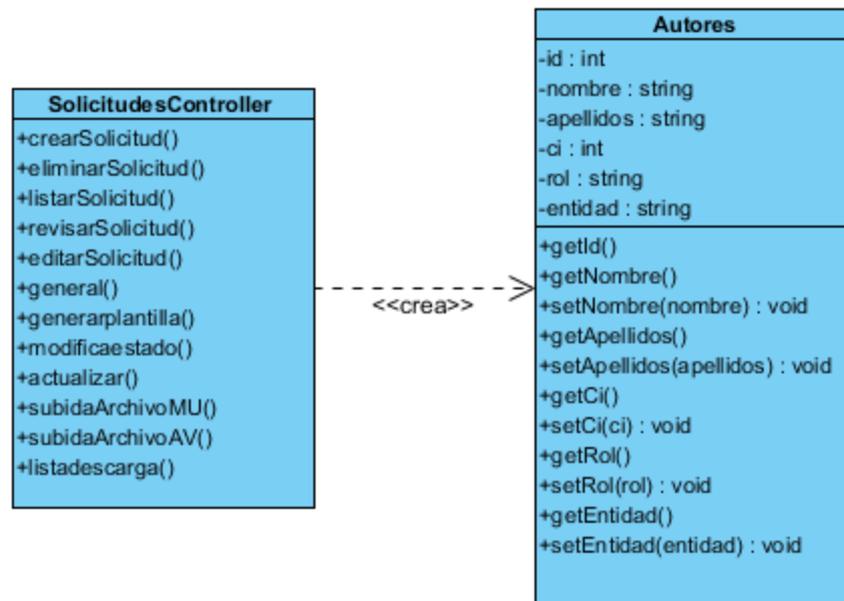


Figura 6: Diagrama de clases que representa el Patrón Creador.

El **patrón Controlador** asigna la responsabilidad de gestionar un mensaje de un evento del sistema a una clase controladora. Sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado.

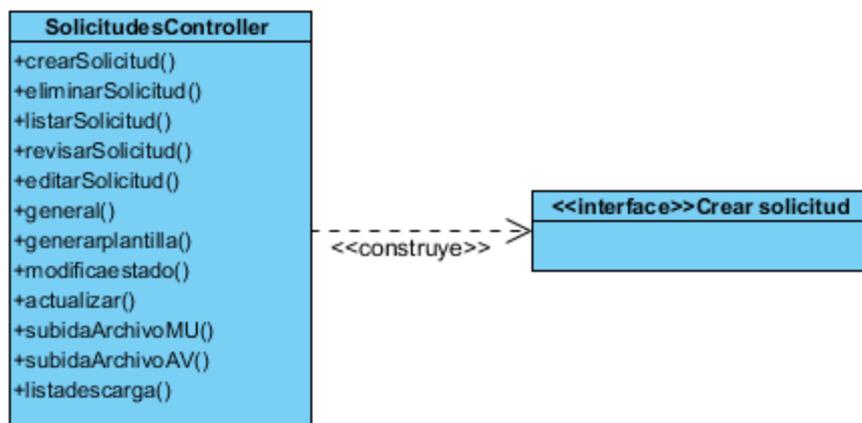


Figura 7: Diagrama de clases que representa el Patrón Controlador.

Además es aplicado el patrón **Bajo acoplamiento**. El mismo establece tener pocos componentes que dependan de otros, pues mientras menos acoplamiento haya más reusable y flexible se vuelve el sistema. En la aplicación las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones

con las de la vista o el controlador, proporcionando esto que la dependencia en este caso sea baja.

Se evidencia también el uso del patrón **Alta Cohesión** el cual plantea que la información que almacena una clase debe de ser coherente y debe estar en la medida de lo posible relacionada con la clase. Para la implementación de la aplicación se hizo uso de las formas de organización que brinda Symfony. El mismo permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello es la clase Actions, la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las properties.

2.9. Tarjetas CRC

“Las tarjetas CRC (Clase, Responsabilidad y Colaboración) son fichas, una por cada clase, en las que se escriben brevemente las responsabilidades de la clase y una lista de los objetos con los que colabora para llevar a cabo esas responsabilidades. La información recopilada se puede enriquecer utilizando diagramas de clases y de interacción. Lo importante no son las tarjetas o los diagramas, sino tener presente la asignación de responsabilidades” (32). A continuación se detallan las clases del módulo mediante estas tarjetas.

Tabla 20: Tarjeta CRC RolesController.

Tarjeta CRC	
Clase: RolesController	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Controlar la interfaz de la sección de roles. • Listar usuarios autenticados en la aplicación. • Asignar roles a los usuarios autenticados. • Crear rol. • Modificar rol. • Eliminar rol. • Eliminar usuario. 	Con la entidad Rol y User tiene relación de asociación.

Tabla 21: Tarjeta CRC SolicitudesController.

Tarjeta CRC	
Clase: SolicitudesController	
Responsabilidad	Colaboración

<ul style="list-style-type: none"> • Controlar la interfaz de la sección de registro de solicitud. • Controla el almacenamiento de solicitudes y autores con derecho sobre la aplicación en la base de datos. • Modificar las solicitudes registradas. • Eliminar las solicitudes registradas. • Mostrar las solicitudes registradas en el sistema. • Subida de documentos asociados con las solicitudes. • Descargar las documentaciones asignadas para cada solicitud. • Generar documentos en formato pdf de las solicitudes registradas. 	<p>Con todas las entidades la relación es de asociación (Rol, User, Autores, Solicitudes, ArchivoMU, ArchivoAV).</p>
--	--

Tabla 22: Tarjeta CRC SeguridadController.

Tarjeta CRC	
Clase: SeguridadController	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Autenticar usuarios a la aplicación (Iniciar sesión, Cerrar sesión). • Verificación de usuarios en el dominio UCI. • Almacenar en las respectivas entidades los usuarios autenticados. 	<p>Con la entidad User y la clase Ldap tiene relación de asociación.</p>

Tabla 23: Tarjeta CRC Ldap.

Tarjeta CRC	
Clase: Ldap	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Verificar que los usuarios que se autenticuen estén en el dominio. • Controlar los datos de los usuarios en el dominio. 	<p>Con la clase SeguridadController de tipo asociación</p>

Tabla 24: Tarjeta CRC Rol.

Tarjeta CRC	
Clase: Rol	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> Manipular la información de la base de datos relacionada con los roles del sistema (rol). 	<p>Con las clases RolesController y SolicitudesController tiene una relación de asociación.</p> <p>Con la entidad User tiene multiplicidad de 1 a 1...* y relación de composición.</p>

Tabla 25: Tarjeta CRC User.

Tarjeta CRC	
Clase: User	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> Manipular la información de la base de datos relacionada con los usuarios del sistema (usuario, correo, nombre, apellidos, rol asignado, contraseña). 	<p>Con la clase RolesController tiene relación de asociación.</p> <p>Con la entidad Rol tiene multiplicidad de 1...* a 1 y relación de composición.</p>

Tabla 26: Tarjeta CRC Autores.

Tarjeta CRC	
Clase: Autores	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> Manipular la información de la base de datos relacionada con los autores del sistema (nombre, apellidos, carnet de identidad, rol, entidad). 	<p>Con la clase SolicitudesController tiene relación de asociación.</p> <p>Con la entidad Solicitudes tiene multiplicidad de 1...* a 1...* y relación de composición.</p>

Tabla 27: Tarjeta CRC Solicitudes.

Tarjeta CRC	
Clase: Solicitudes	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> Manipular la información de la base de datos relacionada con las solicitudes registradas (producto, proyecto, versión, clasificación, tipo) 	<p>Con la clase SolicitudesController tiene relación de asociación.</p>

de registro, centro, cliente, organismo, estado, facultad, solicitante, usuario, cargo, fecha, resumen, datos de autores).	Con la entidad Autores tiene multiplicidad de 1...* a 1...* y con las entidades ArchivoMU y ArchivoAV de 1 a 1. Tiene relación de composición con todas ellas.
--	--

Tabla 28: Tarjeta CRC ArchivoMU.

Tarjeta CRC	
Clase: ArchivoMU	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Subida de archivos relacionados con los manuales de usuario. • Obtener el nombre del archivo subido. • Definir ruta de descarga del archivo. 	<p>Con la clase SolicitudesController tiene relación de asociación.</p> <p>Con la entidad Solicitudes tiene una multiplicidad de 1 a 1 y relación de composición.</p>

Tabla 29: Tarjeta CRC ArchivoAV.

Tarjeta CRC	
Clase: ArchivoAV	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Subida de archivos relacionados con las actas de validación. • Obtener el nombre del archivo subido. • Definir ruta de descarga del archivo. 	<p>Con la clase SolicitudesController tiene relación de asociación.</p> <p>Con la entidad Solicitudes tiene una multiplicidad de 1 a 1 y relación de composición.</p>

2.9. Diagrama de clase

“Un diagrama de clases del diseño representa las especificaciones de las clases e interfaces del software” (32). Este diagrama no es necesario según la metodología XP pero sirve de apoyo en la comprensión de la estructura del sistema. A continuación se muestra el diagrama de clases del diseño del sistema a desarrollar.

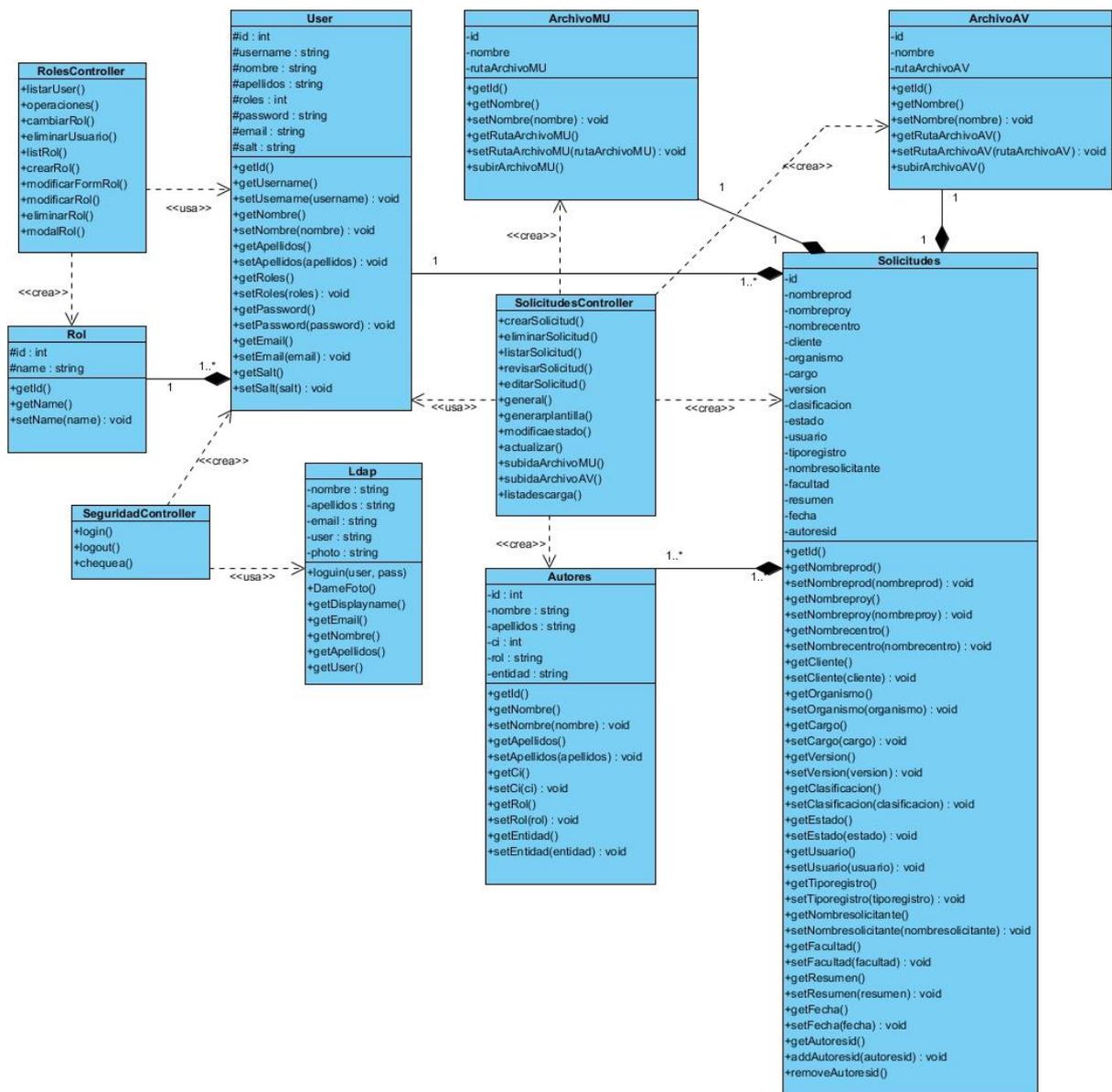


Figura 8: Diagrama de clases del sistema.

2.10. Modelo de datos

Se implementó una base de datos que recoge toda la información necesaria. La misma está compuesta por las tablas: Solicitudes, Autores, ArchivoMU, ArchivoAV, Rol y User. Las relaciones que se establecen permiten hacer las consultas y manipular toda la información referente a solicitudes, los autores y los documentos que pertenecen a cada solicitud, los roles establecidos para cada usuario y los datos de los usuarios que se autentican en el sistema. La entidad Solicitudes almacena toda la información referente a una solicitud creada por un autor determinado. (Ver Figura 10). Por otra parte la entidad Autores contiene datos sobre los autores que tienen derecho sobre una solicitud, dígame nombre, apellidos, carnet de

identidad, rol y entidad. User, a su vez, tiene información referente a cada usuario que es autenticado en el sistema, mientras que Rol contiene los roles que fueron definidos para la aplicación.

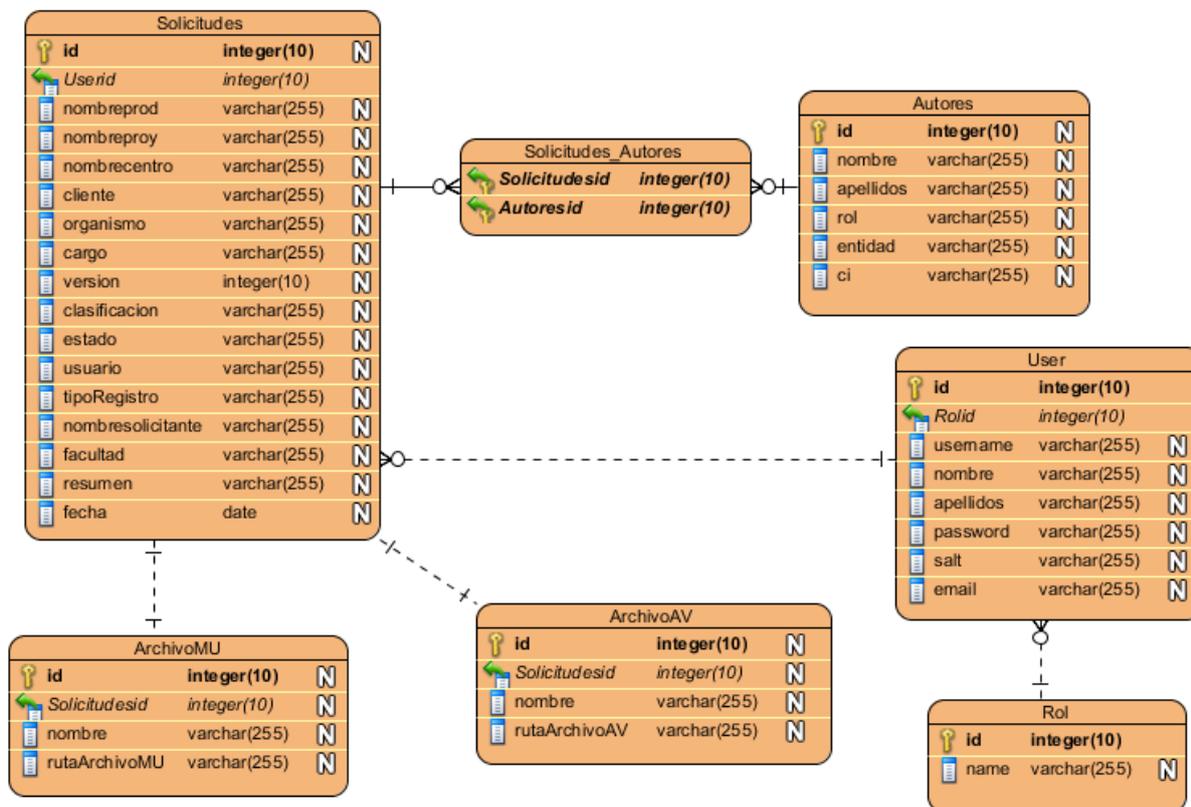


Figura 9: Modelo de datos.

Consideraciones Parciales

- ❖ Los requisitos del sistema y la arquitectura propuesta permitieron obtener un diseño robusto de la aplicación.
- ❖ Las definiciones de historias de usuario permitieron tener la base para la implementación de la aplicación.
- ❖ Se aplicaron los patrones necesarios para lograr un diseño flexible y eficiente.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se procede a realizar la implementación del sistema y al concluirlo se tendrá la aplicación web implementada y probada. Primeramente se realiza la descripción de las tareas de ingeniería por cada historia de usuario. Además se aplican las pruebas pertinentes para la validación del correcto funcionamiento del sistema. Por otra parte se muestra el diagrama de despliegue del sistema a implementar.

3.1. Fase de Implementación

En esta fase se realiza la implementación de las historias de usuario correspondientes a cada iteración, chequeando el plan de iteraciones por si hay que realizar modificaciones y se crean las tareas de programación para implementar exitosamente cada HU. A continuación se detallan las tareas a desarrollar para cada HU.

3.1.1. Iteración 1

Tabla 30: Tareas por HU de la iteración 1.

Número	Historias de usuario	Tareas por Historias de Usuario
1	Autenticar usuario	<ol style="list-style-type: none">1. Implementar el formulario encargado de generar la autenticación al sistema.2. Adjuntar a las clases implementadas la clase de verificación haciendo uso de LDAP.3. Generar las entidades Rol y User en la base de datos de la aplicación.
2	Insertar solicitud	<ol style="list-style-type: none">1. Implementar el formulario encargado de la inserción de solicitudes en el sistema.2. Generar las entidades Solicitudes y Autores en la base de datos de la aplicación.3. Validar los campos por los que está compuesto el formulario.
3	Modificar solicitud	<ol style="list-style-type: none">1. Mostrar las solicitudes almacenadas en la base de datos.2. Almacenar los cambios en las entidades pertinentes (Solicitudes, Autores, SolicitudAutor).4. Validar los campos por los que está compuesto el formulario.

4	Revisar solicitud	<ol style="list-style-type: none"> 1. Mostrar las solicitudes almacenada en la base de datos. (Ver la tabla 29) 2. Actualizar los cambios en las entidades pertinentes (Solicitudes, Autores, SolicitudAutor I). (Ver la Tabla 30) 3. Generar descarga y subida de ficheros. <ol style="list-style-type: none"> 1. Exportar formularios a formatos pdf.
---	-------------------	---

Tabla 31: Tarea 1 de la HU 1.

Tarea de Ingeniería	
Número de tarea: 1	Número de Historia de Usuario: 1
Nombre de la tarea: Implementar el formulario encargado de generar la autenticación al sistema.	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 6/2/2014	Fecha Fin: 7/2/2014
Descripción: Se diseña el formulario con los campos necesarios para la autenticación (usuario y contraseña).	

Tabla 32: Tarea 2 de la HU 1.

Tarea de Ingeniería	
Número de tarea: 2	Número de Historia de Usuario: 1
Nombre de la tarea: Adjuntar a las clases implementadas la clase de verificación haciendo uso de LDAP.	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.1
Fecha Inicio: 10/2/2014	Fecha Fin: 12/2/2014
Descripción: Se adjunta a las clases del sistema, la clase Ldap, para la verificación haciendo uso del Protocolo Ligero de Acceso a Directorios.	

Tabla 33: Tarea 3 de la HU 1.

Tarea de Ingeniería	
Número de tarea: 3	Número de Historia de Usuario: 1
Nombre de la tarea: Generar las entidades Rol y User en la base de datos de la aplicación.	

Tipo de tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 13/2/2014	Fecha Fin: 14/2/2014
Descripción: Se implementan las entidades Rol y User en la base de datos de la aplicación para controlar los usuarios y roles en el sistema.	

Tabla 34: Tarea 1 de la HU 2.

Tarea de Ingeniería	
Número de tarea: 1	Número de Historia de Usuario: 2
Nombre de la tarea: Implementar el formulario encargado de la inserción de solicitudes en el sistema.	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.3
Fecha Inicio: 17/2/2014	Fecha Fin: 18/2/2014
Descripción: Se diseña el formulario para la inserción de solicitudes, teniendo en cuenta todos los campos que debe llevar la misma.	

Tabla 35: Tarea 2 de la HU 2.

Tarea de Ingeniería	
Número de tarea: 2	Número de Historia de Usuario: 2
Nombre de la tarea: Generar las entidades Solicitudes y Autores en la base de datos de la aplicación.	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.3
Fecha Inicio: 19/2/2014	Fecha Fin: 21/2/2014
Descripción: Se crean las entidades Solicitudes, Autores y SolicitudAutor para el almacenamiento de toda la información que recoge el formulario según las características de la solicitud.	

Tabla 36: Tarea 3 de la HU 2.

Tarea de Ingeniería	
Número de tarea: 3	Número de Historia de Usuario: 2
Nombre de la tarea: Validar los campos por los que está compuesto el formulario.	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.3
Fecha Inicio: 21/2/2014	Fecha Fin: 21/2/2014
Descripción: Se validan los campos correspondientes del formulario según el tipo de dato que soporta cada elemento de la solicitud.	

Tabla 37: Tarea 1 de la HU 3.

Tarea de Ingeniería	
Número de tarea: 1	Número de Historia de Usuario: 3
Nombre de la tarea: Mostrar las solicitudes almacenadas en la base de datos.	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.1
Fecha Inicio: 3/3/2014	Fecha Fin: 4/3/2014
Descripción: Se muestran las solicitudes almacenadas en la base de datos de la aplicación.	

Tabla 38: Tarea 2 de la HU 3.

Tarea de Ingeniería	
Número de tarea: 2	Número de Historia de Usuario: 3
Nombre de la tarea: Almacenar los cambios en las entidades pertinentes (Solicitudes, Autores, SolicitudAutor).	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 5/3/2014	Fecha Fin: 6/3/2014
Descripción: Se guardan los cambios realizados en las entidades mencionadas anteriormente.	

Tabla 39: Tarea 3 de la HU 3.

Tarea de Ingeniería	
Número de tarea: 3	Número de Historia de Usuario: 3
Nombre de la tarea: Validar los campos por los que está compuesto el formulario.	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 7/3/2014	Fecha Fin: 7/3/2014
Descripción: Se validan los campos correspondientes del formulario según el tipo de dato que soporta cada elemento de la solicitud.	

Tabla 40: Tarea 3 de la HU 4.

Tarea de Ingeniería	
Número de tarea: 3	Número de Historia de Usuario: 4
Nombre de la tarea: Generar descarga y subida de ficheros.	

Tipo de tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 13/3/2014	Fecha Fin: 18/3/2014
Descripción: Se implementa los componentes para la subida y descarga de archivos anexados a las solicitudes.	

Tabla 41: Tarea 4 de la HU 4.

Tarea de Ingeniería	
Número de tarea: 4	Número de Historia de Usuario: 4
Nombre de la tarea: Exportar formularios a formatos pdf.	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 19/3/2014	Fecha Fin: 21/3/2014
Descripción: Generar documentos en formatos pdf a partir de la información adjuntada al formulario de solicitudes.	

3.1.2. Iteración 3

Tabla 42: Tareas por HU de la iteración 2.

Número	Historias de usuario	Tareas por Historias de Usuario
5	Asignar rol	<ol style="list-style-type: none"> 1. Mostrar usuarios almacenados de la entidad User. 2. Generar un formulario para el control de roles. 3. Almacenar los cambios en las entidades pertinentes (User y Rol).
6	Listar solicitud	<ol style="list-style-type: none"> 1. Mostrar las solicitudes almacenada en la base de datos. (Ver tabla 37) 2. Generar una estructura de tabla para visualizar los elementos de cada solicitud.
7	Enviar notificación	<ol style="list-style-type: none"> 1. Incluir la librería Swiftmailer en el framework para el envío de correos.
8	Eliminar solicitud	<ol style="list-style-type: none"> 1. Mostrar las solicitudes almacenada en la base de datos. (Ver tabla 37)

Tabla 43: Tarea 1 de la HU 5.

Tarea de Ingeniería	
Número de tarea: 1	Número de Historia de Usuario: 5
Nombre de la tarea: Mostrar usuarios almacenados de la entidad User.	

Tipo de tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 24/2/2014	Fecha Fin: 25/2/2014
Descripción: Se muestran los usuarios existentes en la entidad User.	

Tabla 44: Tarea 2 de la HU 5.

Tarea de Ingeniería	
Número de tarea: 2	Número de Historia de Usuario: 5
Nombre de la tarea: Generar un formulario para el control de roles.	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 26/2/2014	Fecha Fin: 27/2/2014
Descripción: Se implementa un formulario para controlar la asignación de roles por parte del administrador.	

Tabla 45: Tarea 3 de la HU 5

Tarea de Ingeniería	
Número de tarea: 3	Número de Historia de Usuario: 5
Nombre de la tarea: Almacenar los cambios en las entidades pertinentes (User y Rol).	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.1
Fecha Inicio: 28/2/2014	Fecha Fin: 28/2/2014
Descripción: Se almacenan los cambios pertinentes en dichas entidades.	

Tabla 46: Tarea 2 de la HU 6.

Tarea de Ingeniería	
Número de tarea: 2	Número de Historia de Usuario: 6
Nombre de la tarea: Generar una estructura de tabla para visualizar los elementos de cada solicitud.	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 10/3/2014	Fecha Fin: 12/3/2014
Descripción: Se validan los campos correspondientes del formulario según el tipo de dato que soporta cada elemento de la solicitud.	

Tabla 47: Tarea 1 de la HU 7.

Tarea de Ingeniería	
Número de tarea: 4	Número de Historia de Usuario: 6
Nombre de la tarea: Incluir la librería Swiftmailer en el framework para el envío de correos.	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.2
Fecha Inicio: 24/3/2014	Fecha Fin: 25/3/2014
Descripción: Hacer uso de la librería Swiftmailer en Symfony para el envío de correos de forma automática.	

3.2. Diagrama de despliegue

“El despliegue provee un modelo detallado de la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema. Muestran la disposición física de los distintos nodos que componen un sistema y como están repartidos los componentes sobre dichos nodos. Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional, como son los procesadores o dispositivos de hardware, existen relaciones que representan medios de comunicación entre ellos”. (33)

A continuación se muestra del diagrama de despliegue que se corresponde con el sistema que se desea implementar.

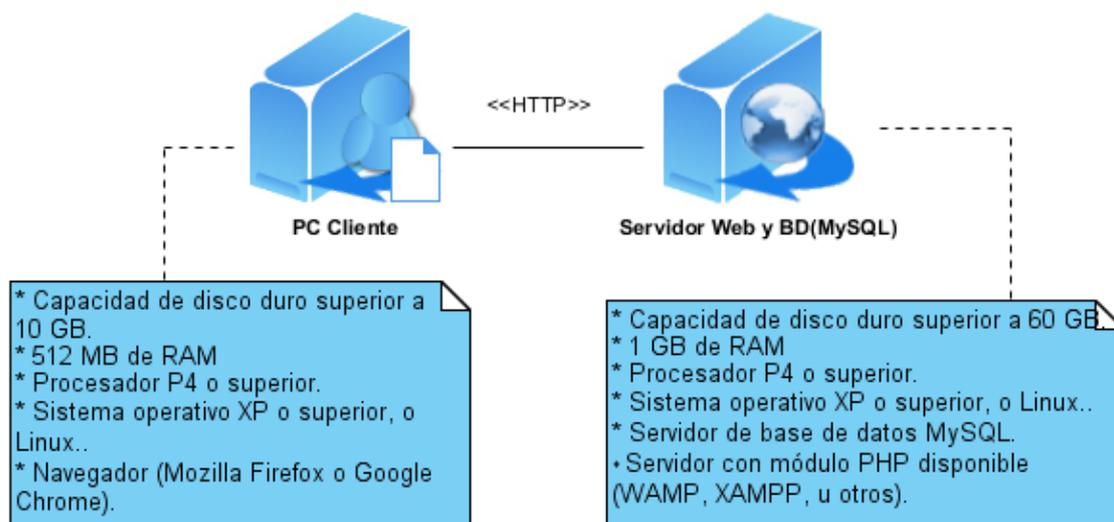


Figura 10: Diagrama de despliegue.

- ❖ **PC Cliente:** Serán las estaciones de trabajo que el usuario utilizará para acceder a la aplicación Web.
- ❖ **Servidor:** En este nodo están contemplados los servidores web y de base de datos de la aplicación.
 - **Servidor web:** El servidor de aplicación es utilizado para la publicación de la aplicación. Es la herramienta principal para ejecutar la lógica de negocio en el lado del servidor y es el encargado de ejecutar el código de las páginas servidor. En el diagrama de despliegue se utiliza el servidor de aplicación Wamp.
 - **Servidor de Base de Datos:** Se hace referencia al gestor de bases de datos donde se encuentran los datos necesarios para el trabajo con el sistema. El servidor de base de datos elegido es MySQL, que es un gestor de licencia no propietaria o sea libre y muy potente.

Conexión HTTP: Es el protocolo utilizado entre los navegadores de los clientes y el servidor Web. Este elemento de la arquitectura representa un tipo de comunicación no orientado a la conexión entre clientes y servidor.

3.3. Diagrama de componentes

Este tipo de diagramas representa cómo un sistema de software es dividido en componentes, que pueden ser archivos, módulos, bibliotecas, ejecutables o paquetes y muestra las dependencias entre ellos.

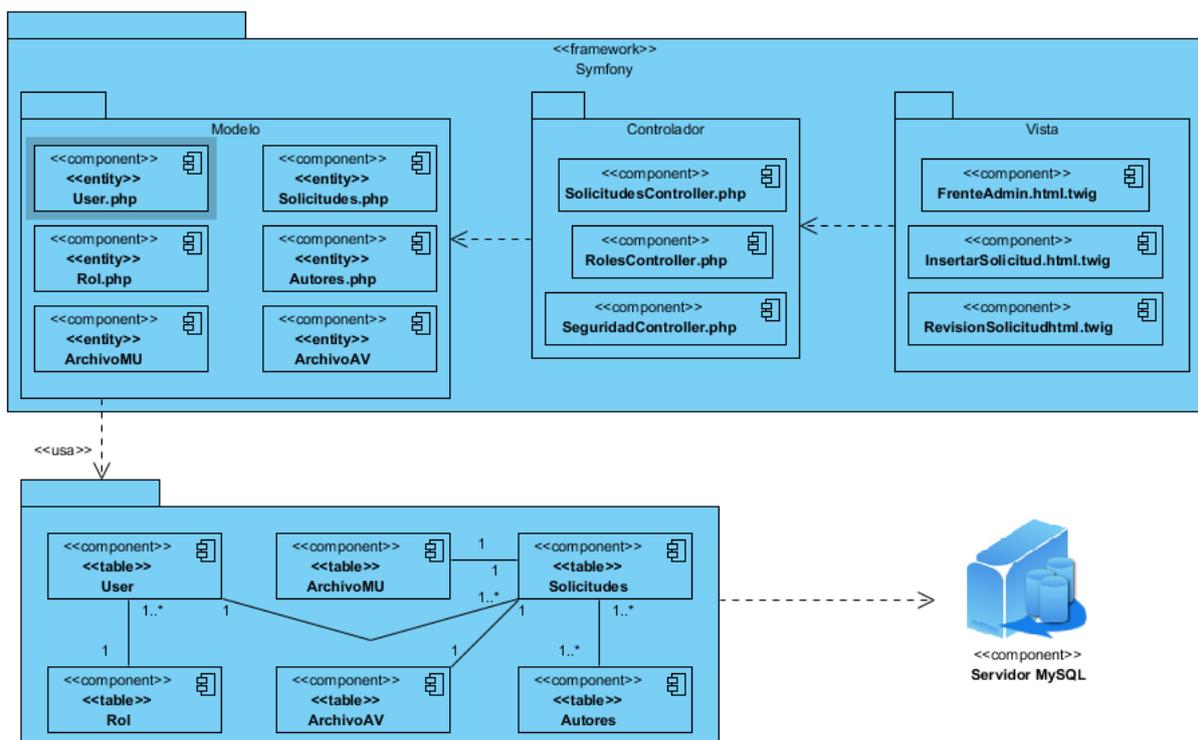


Figura 11: Diagrama de componentes.

4.3. Casos de pruebas de aceptación

Con XP se realizan las pruebas unitarias y las pruebas de aceptación. Las primeras son desarrolladas por los programadores para verificar que el código funciona, mientras que las segundas son utilizadas para evaluar si al final de la iteración se obtuvo la funcionalidad deseada por el cliente.

Las pruebas de aceptación también son conocidas como pruebas de caja negra, pues es el propio cliente quien las realiza en compañía de uno de los representantes del equipo de desarrollo y se orientan a las funcionalidades del sistema. Su objetivo es comprobar, desde la perspectiva del usuario final, el cumplimiento de las especificaciones definidas para el producto. Una HU solo se considerará terminada cuando haya pasado correctamente todas las pruebas de aceptación.

A continuación se muestran las pruebas de aceptación realizadas a la solución propuesta:

Tabla 48: Prueba de aceptación para la HU "Autenticar usuario"

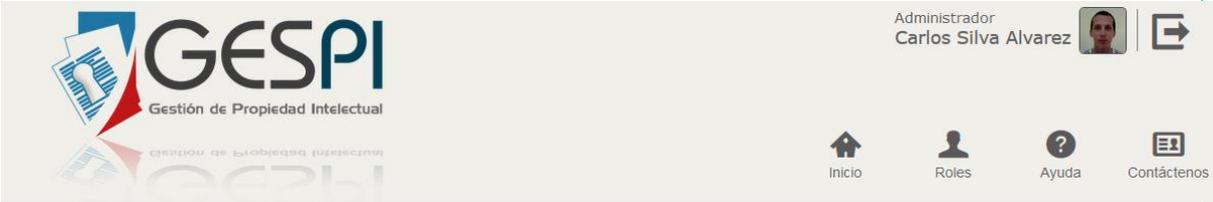
Caso de Prueba de Aceptación	
Código: H1_P1	Historia de Usuario: 1
Nombre: Autenticar usuario	
Descripción: El sistema debe permitir la autenticación de usuario para acceder al entorno de trabajo del mismo haciendo uso de usuario y contraseña.	
Condiciones de ejecución: La página de autenticación de la aplicación debe estar abierta.	
Entrada/Pasos de ejecución: <ol style="list-style-type: none"> 1. Introducir los datos en los campos (usuario y contraseña). 2. Click en el botón Iniciar sesión. 	
Resultado esperado: El usuario debe ser autenticado en el sistema. El mismo debe ser mostrado en la parte superior derecha.	
Resultado obtenido:	
 <p>The screenshot shows the top navigation bar of the GESPI application. On the left is the logo for 'GESPI Gestión de Propiedad Intelectual'. On the right, the user 'Administrador Carlos Silva Alvarez' is logged in, with a profile picture and a logout icon. Below the user name are four navigation icons: 'Inicio' (Home), 'Roles', 'Ayuda' (Help), and 'Contáctenos' (Contact Us).</p>	
Evaluación de la prueba: Satisfactoria.	

Tabla 49: Prueba de aceptación para la HU "Insertar solicitud"

Caso de Prueba de Aceptación	
Código: H2_P2	Historia de Usuario: 2

Nombre: Insertar solicitud.
Descripción: El usuario puede realizar solicitudes para la certificación de un producto.
Condiciones de ejecución: El usuario debe estar autenticado en la aplicación.
Entrada/Pasos de ejecución: <ol style="list-style-type: none"> 1. Click en el botón Registrar del módulo Registro de Solicitud ubicado en la parte izquierda del menú principal. 2. Introducir los datos correspondientes a la solicitud. 3. Click en el botón Crear solicitud. 4. Subir la documentación asociada a la solicitud (Manual de Usuario y Acta de Validación).
Resultado esperado: El sistema debe almacenar en la base de datos los valores introducidos en la solicitud. La documentación debe ser guardada en el directorio de archivos del sistema. Muestra un mensaje de notificación con el resultado obtenido.
Resultado obtenido: <div style="background-color: #0099cc; color: white; padding: 10px; border: 1px solid #ccc;"> <p>Su solicitud ha sido creada satisfactoriamente</p> <div style="text-align: right;"> Cerrar </div> </div>
Evaluación de la prueba: Satisfactoria.

Tabla 50: Prueba de aceptación para la HU "Modificar solicitud"

Caso de Prueba de Aceptación	
Código: H3_P3	Historia de Usuario: 3
Nombre: Modificar solicitud.	

Descripción: El usuario puede modificar las solicitudes de registro que fueron creadas por él.

Condiciones de ejecución: El usuario debe estar autenticado en la aplicación y debe existir la solicitud a modificar en la base de datos de la aplicación.

Entrada/Pasos de ejecución:

1. Click en el botón **Listar** del módulo **Solicitudes Registradas** ubicado en la parte central del menú principal.
2. Click en el botón **Modificar** asociado a la solicitud a editar.
3. Modificar los datos pertinentes.
4. Click en el botón **Modificar**.

Resultado esperado: El sistema debe actualizar en la base de datos los valores introducidos en la solicitud. Se debe mostrar un mensaje de notificación con el resultado obtenido.

Resultado obtenido:

Su solicitud ha sido modificada satisfactoriamente

Cerrar

Evaluación de la prueba: Satisfactoria.

Tabla 51: Prueba de aceptación para la HU "Listar solicitud"

Caso de Prueba de Aceptación																																					
Código: H4_P4	Historia de Usuario: 4																																				
Nombre: Listar solicitud.																																					
Descripción: El sistema debe listar las solicitudes creadas por parte del usuario.																																					
Condiciones de ejecución: El usuario debe estar autenticado en la aplicación y debe existir la solicitud en la base de datos de la aplicación.																																					
Entrada/Pasos de ejecución: <ol style="list-style-type: none">1. Click en el botón Listar del módulo Solicitudes Registradas ubicado en la parte central del menú principal.																																					
Resultado esperado: Las solicitudes deben ser listadas en la aplicación.																																					
Resultado obtenido:																																					
<table border="1"><thead><tr><th>PRODUCTO</th><th>TIPO DE REGISTRO</th><th>USUARIO</th><th>FECHA</th><th>ESTADO</th><th></th><th></th><th></th><th></th></tr></thead><tbody><tr><td>SCADA-Almiqui</td><td>Software</td><td>betty</td><td>05-06-2014</td><td>Iniciado</td><td></td><td></td><td></td><td></td></tr><tr><td>Sistema de Análisis Petrofísico</td><td>Software</td><td>yantunez</td><td>01-06-2014</td><td>Iniciado</td><td></td><td></td><td></td><td></td></tr><tr><td>Sistema de Entrenamientos de Indicios Aduaneros</td><td>Software</td><td>rmferrer</td><td>28-04-2014</td><td>Iniciado</td><td></td><td></td><td></td><td></td></tr></tbody></table>		PRODUCTO	TIPO DE REGISTRO	USUARIO	FECHA	ESTADO					SCADA-Almiqui	Software	betty	05-06-2014	Iniciado					Sistema de Análisis Petrofísico	Software	yantunez	01-06-2014	Iniciado					Sistema de Entrenamientos de Indicios Aduaneros	Software	rmferrer	28-04-2014	Iniciado				
PRODUCTO	TIPO DE REGISTRO	USUARIO	FECHA	ESTADO																																	
SCADA-Almiqui	Software	betty	05-06-2014	Iniciado																																	
Sistema de Análisis Petrofísico	Software	yantunez	01-06-2014	Iniciado																																	
Sistema de Entrenamientos de Indicios Aduaneros	Software	rmferrer	28-04-2014	Iniciado																																	
<p>Prev 1 Next</p>																																					
Evaluación de la prueba: Satisfactoria.																																					

Tabla 52: Prueba de aceptación para la HU "Enviar notificación"

Caso de Prueba de Aceptación	
Código: H6_P6	Historia de Usuario: 6
Nombre: Enviar notificación.	
Descripción: El sistema debe enviar notificaciones de forma automática al autor de la solicitud, informando el estado en que se encuentra la misma.	
Condiciones de ejecución: El estado de una solicitud debe ser modificada.	
Entrada/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Click en el botón Revisar del módulo Revisión de Solicitud ubicado en la parte derecha del menú principal. 2. Click en el botón Modificar solicitud para editar el estado de la solicitud a revisar. 	
Resultado esperado: El sistema debe enviar un correo de notificación al autor de la solicitud.	
Resultado obtenido: La notificación fue enviada por correo al autor de la solicitud	
Evaluación de la prueba: Satisfactoria.	

Tabla 53: Prueba de aceptación para la HU "Revisar solicitud"

Caso de Prueba de Aceptación	
Código: H5_P5	Historia de Usuario: 5
Nombre: Revisar solicitud.	

Descripción: El sistema debe listar las solicitudes creadas por parte del usuario, posibilitando la descarga de documentación y modificación del estado de las solicitudes.

Condiciones de ejecución: El Asesor debe estar autenticado en la aplicación y debe existir solicitudes para iniciar el proceso de revisión.

Entrada/Pasos de ejecución:

1. Click en el botón **Revisar** del módulo **Revisión de Solicitud** ubicado en la parte derecha del menú principal.
2. Click en el botón **Modificar solicitud** para editar el estado de la solicitud a revisar (En revisión).
3. Click en el botón **Descargar** para obtener los documentos asociados a la solicitud (Manual de Ayuda y Acta de Validación).
4. Revisar la documentación descargada por parte de los asesores en busca de inconsistencias.
5. Click en el botón **Adjuntar archivo** para subir los documentos revisados por el asesor.
6. Click en el botón **Modificar solicitud** para editar el estado de la solicitud revisada a (Aceptada).
7. Click en el botón **Exportar solicitud** una vez revisada la solicitud más la documentación para generar un documento con la información de la solicitud registrada.

Resultado esperado: Los documentos deben ser descargados a algún directorio según las preferencias del usuario. Además el estado de la solicitud debe ser actualizado en la base de datos. Por otra parte deben ser subidos los documentos revisados. Por último, se debe generar un documento donde recoja todos los datos relacionados con la solicitud.

Resultado obtenido: Fue generado un documento en formato pdf. Además el estado de la solicitud fue actualizado y los documentos fueron descargados y subidos una vez revisados.

Evaluación de la prueba: Satisfactoria.

Tabla 54: Prueba de aceptación para la HU " Asignar rol "

Caso de Prueba de Aceptación	
Código: H7_P7	Historia de Usuario: 7
Nombre: Asignar rol.	

Descripción: El administrador podrá Asignar roles a usuarios en el sistema.
Condiciones de ejecución: El administrador debe estar autenticado en la aplicación. Además, deben existir usuarios registrados en la aplicación.
Entrada/Pasos de ejecución: <ol style="list-style-type: none"> 1. Click en el botón Roles ubicado en la parte superior derecha de la aplicación. 2. Modificar el rol del usuario haciendo uso del menú desplegable en el campo Rol. 3. Click en el botón Cambiar rol.
Resultado esperado: Los roles asignados deben ser actualizados en la base de datos del sistema. Además en la página deben mostrarse los cambios realizados.
Resultado obtenido: El rol del usuario fue actualizado en la base de datos de la aplicación.
Evaluación de la prueba: Satisfactoria.

Tabla 55: Prueba de aceptación para la HU " Eliminar solicitud "

Caso de Prueba de Aceptación	
Código: H8_P8	Historia de Usuario: 8
Nombre: Eliminar solicitud.	
Descripción: El autor podrá eliminar las solicitudes que fueron registradas por él.	
Condiciones de ejecución: El autor debe estar autenticado en la aplicación. Además, deben existir solicitudes registradas por él.	
Entrada/Pasos de ejecución: <ol style="list-style-type: none"> 1. Click en el botón Listar del módulo Solicitudes Registradas ubicado en la parte central del menú principal. 2. Click en el botón Eliminar asociado a la solicitud a ser eliminada. 3. Aceptar el mensaje de confirmación de eliminación. 	
Resultado esperado: La solicitud debe ser eliminada de la aplicación.	
Resultado obtenido:	
	
Evaluación de la prueba: Satisfactoria.	

Tabla 56: Prueba de aceptación para la HU " Eliminar solicitud "

Caso de Prueba de Aceptación	
Código: H9_P9	Historia de Usuario: 9
Nombre: Eliminar usuario.	

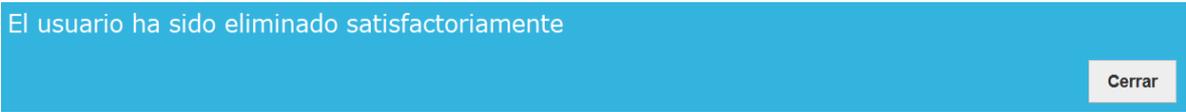
Descripción: El administrador podrá eliminar los usuarios que han sido registradas en la aplicación.
Condiciones de ejecución: El administrador debe estar autenticado en la aplicación. Además, deben existir usuarios registrados en la aplicación.
Entrada/Pasos de ejecución: <ol style="list-style-type: none"> 1. Click en el botón Roles ubicado en la parte superior derecha de la aplicación. 2. Click en el botón Eliminar usuario asociado al usuario a eliminar. 3. Aceptar el mensaje de confirmación de eliminación.
Resultado esperado: La solicitud debe ser eliminada de la aplicación. Se muestra un mensaje de notificación con el resultado de la acción.
Resultado obtenido: 
Evaluación de la prueba: Satisfactoria.

Tabla 57: Prueba de aceptación para la HU " Crear rol "

Caso de Prueba de Aceptación	
Código: H10_P10	Historia de Usuario: 10
Nombre: Crear rol.	
Descripción: El administrador podrá crear roles en la aplicación.	
Condiciones de ejecución: El administrador debe estar autenticado en la aplicación.	
Entrada/Pasos de ejecución: <ol style="list-style-type: none"> 1. Click en el botón Roles ubicado en la parte superior derecha de la aplicación. 2. Click en el botón Gestionar Rol. 3. Click en el botón Crear rol. 4. Crear el rol. 5. Click en el botón Crear rol. 	
Resultado esperado: La aplicación lista los usuarios registrados para asignar el rol creado. El rol es creado en la aplicación.	
Resultado obtenido: El fue creado en la aplicación.	
Evaluación de la prueba: Satisfactoria.	

Tabla 58: Prueba de aceptación para la HU " Modificar rol "

Caso de Prueba de Aceptación	
Código: H11_P11	Historia de Usuario: 11
Nombre: Modificar rol.	

Descripción: El administrador podrá modificar roles en la aplicación.
Condiciones de ejecución: El administrador debe estar autenticado en la aplicación.
Entrada/Pasos de ejecución: <ol style="list-style-type: none"> 1. Click en el botón Roles ubicado en la parte superior derecha de la aplicación. 2. Click en el botón Gestionar Rol. 3. Click en el botón Modificar rol. 4. Modificar el rol. 5. Click en el botón Modificar rol.
Resultado esperado: La aplicación lista los roles registrados con los cambios realizados. Actualiza la base de datos de la aplicación
Resultado obtenido: <div style="text-align: center;"> <p>Creación de Roles</p> <hr/> <p>Nombre del Rol</p> <input type="text" value="Autor"/> <input type="button" value="Modificar rol"/> </div>
Evaluación de la prueba: Satisfactoria.

Tabla 59: Prueba de aceptación para la HU " Eliminar rol "

Caso de Prueba de Aceptación	
Código: H12_P11	Historia de Usuario: 12
Nombre: Eliminar rol.	
Descripción: El administrador podrá eliminar roles en la aplicación.	
Condiciones de ejecución: El administrador debe estar autenticado en la aplicación.	
Entrada/Pasos de ejecución: <ol style="list-style-type: none"> 1. Click en el botón Roles ubicado en la parte superior derecha de la aplicación. 2. Click en el botón Gestionar Rol. 3. Click en el botón Eliminar rol. 4. Aceptar mensaje de confirmación. 5. De existir un usuario con el rol a eliminar: <ol style="list-style-type: none"> a. Seleccionar un rol para asignar al usuario. b. Click en el botón Asignar rol. 	
Resultado esperado: La aplicación lista los roles registrados, exceptuando el rol eliminado.	
Resultado obtenido: El rol fue eliminado de la aplicación	
Evaluación de la prueba: Satisfactoria.	

3.3. Resultados

La solución del problema planteado ayudará a resolver los incidentes que existían en el centro en cuanto al proceso de registro de la propiedad intelectual de sus productos. Por una parte los registros que se hacían manualmente, ahora se podrán realizar a través de una aplicación web quedando estas registradas en la base de datos de la aplicación. Por otro lado la documentación que genera el registro de una solicitud, será adjuntada de manera online a ella, previendo que esta no sea intercambiada por correo electrónico.

Además los especialistas encargados de llevar a cabo la revisión no tendrán que validarla mediante correo electrónico, pues la aplicación constará con un módulo de revisión el cual les facilitará el trabajo. El autor conocerá sobre el estado en que se encuentre su solicitud mediante un sistema de notificación por vía correo.

Consideraciones Parciales

- ❖ Fueron definidas las tareas de ingeniería, las cuales evitaron una sobrecarga de trabajo, agilizando aún más el proceso.
- ❖ La aplicación de las pruebas de aceptación a la solución permitieron comprobar la correcta implementación de las historias de usuarios definidas con anterioridad.

CONCLUSIONES GENERALES

La investigación desarrollada y los resultados obtenidos permitieron arrojar las siguientes conclusiones:

- ❖ El análisis de las características de los Sistemas de Gestión de Propiedad Intelectual de productos informáticos, permitió el diseño de un Sistema de Gestión para la Propiedad Intelectual, que se adecua a las necesidades del Centro de Informática Industrial.
- ❖ La búsqueda y análisis de sistemas semejantes, demostró que las herramientas tanto nacionales como internacionales, poseen características no requeridas por el cliente. Además, los productos internacionales tienen altos costos en los mercados y pertenecen a empresas privadas. Todo esto evidencia la necesidad de crear un nuevo producto de este tipo para el Centro de Informática Industrial.
- ❖ Mediante la metodología XP se logró generar los artefactos correspondientes para la implementación de la aplicación.
- ❖ Se obtuvo una aplicación web funcional (probada durante su proceso de desarrollo y validada utilizando pruebas de aceptación), desarrollada con tecnologías libres y provista de elementos de seguridad, gracias a la utilización del framework Symfony2.

RECOMENDACIONES

Luego de haber culminado la investigación se propone como recomendaciones:

1. Implantar la aplicación desarrollada en el Centro de Informática Industrial, como instrumento para controlar el proceso de registro de propiedad intelectual por el Grupo de Inteligencia Empresarial de dicho centro, después que la misma sea hosteada.
2. Implementar funcionalidades que permitan realizar análisis estadísticos a partir de la información de los registros.
3. Extender la generación automática de formularios en Symfony en todo el sistema.
4. Extender el sistema a otros procesos del Grupo como la Gestión de la Contratación.
5. Hacer extensivo el uso de la aplicación a todos los centros productivos de la universidad.

REFERENCIAS

1. **Real Academia de la Lengua.** *Diccionario de la Lengua Española.* Madrid : Editorial Espasa-Calpe, S.A, 1970.
2. **Robledo, Santiago Márquez.** *Principios generales del Derecho de Autor.* Facultad de Ciencias Jurídicas, Pontificia Universidad Javeriana. Bogotá : s.n., 2004.
3. **Enciclopedia Cubana en la Red.** [En línea] [Citado el: 12 de noviembre de 2013.] http://www.ecured.cu/index.php/Propiedad_intelectual.
4. **Pombert, Ania Torres.** *Diseño de sistemas internos de propiedad intelectual en el modelo cubano: una propuesta metodológica.* La Habana, Cuba : s.n.
5. **(OMPI), Organización Mundial de la Propiedad Intelectual.** *Sitio web de la OMPI.*
6. **Enciclopedia Cubana en la Red.** [En línea] [Citado el: 22 de noviembre de 2013.] http://www.ecured.cu/index.php/Derecho_de_Autor.
7. **Centro Nacional de Derecho de Actor, CENDA.** *El derecho de autor en Cuba.* La Habana : s.n.
8. **Suárez, R. Ramirez Anáili.** *Sistema de Propiedad Intelectual para Centro de Investigación y Desarrollo.*
9. **Enciclopedia Cubana en la Red.** [En línea] [Citado el: 24 de noviembre de 2013.] http://www.ecured.cu/index.php/Metodolog%C3%ADas_de_desarrollo_de_software.
10. **Méndez, Alejandra Virrueta.** *Metodologías de desarrollo de software.* Diciembre 2010.
11. **Ruiz, Beymar Jiménez, y otros.** *Metodología de desarrollo de software MSF.* Facultad de Ingeniería en Ciencias de la Computación y Telecomunicaciones, Universidad Autónoma Gabriel Rene Moreno. Santa Cruz. Bolivia : s.n., 2012.
12. **E. Gallo and M. Vergara.** *Metodologías de Desarrollo de Software Ágiles.* European Software Institute. 2006.
13. **Gracia, J.** *Gestion de Proyectos con SCRUM.* 2006.
14. **Peñalber, G., García, S. y Meneses, A.** *SXP, Metodología Ágil para el desarrollo de software.* La Habana : s.n., 2010.

15. Beck, Kent y Fowler, Martin. *Planning Extreme Programming*. s.l.: Person Education, 2000. pág. 160.
16. Enciclopedia Cubana en la Red. [En línea] [Citado el: 6 de diciembre de 2013.] www.ecured.cu/index.php/Visual_Paradigm.
17. Enciclopedia Cubana en la Red. [En línea] [Citado el: 4 de 12 de 2013.] <http://www.ecured.cu/index.php/Php>.
18. Maestros del Web. [En línea] [Citado el: 6 de 12 de 2013.] <http://www.maestrosdelweb.com/editorial/los-diferentes-lenguajes-de-programacion-para-la-web/>.
19. Instituto Tecnológico de Veracruz. [En línea] [Citado el: 6 de diciembre de 2013.] prograweb.com.mx/pweb/0202ladoCliente.html.
20. Boble, Mark. *Drupal 7 First Look*.
21. Phil Bartle. *INFORMACIÓN PARA LA GESTIÓN y gestión de la información*. 2009.
22. Oracle. [En línea] [Citado el: 13 de 2 de 2014.] <http://www.oracle.com/es/products/mysql/index.html?ssSourceSiteId=null>.
23. Entorno de Desarrollo Integrado. [En línea] [Citado el: 6 de diciembre de 2013.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.
24. NetBeans. [En línea] [Citado el: 6 de diciembre de 2013.] <http://netbeans.org/>.
25. Gutiérrez, Javier J. *¿Qué es un framework web?*
26. *Curso básico de Symfony 2*. 2012.
27. Potencier, Fabien y Zaninotto, François. *Symfony 1.2, la guía definitiva*.
28. Sommerville, Ian. *Ingeniería del Software*. Séptima edición.
29. Eguiluz, Javier. *Desarrollo web ágil con Symfony2*. 2012.
30. Larman, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Mexico : s.n., 1999.
31. Grosso, A. *Patrones GRASP*. 2011.
32. Larman, Craig. *UML y patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. 2006.

- 33. Modelo de Implementación: Diagramas de Componentes y Despliegue. [En línea] [Citado el: 18 de 4 de 2014.]**
<http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.
- 34. Soler, Otto Batista, y otros.** *Registro de la Propiedad Intelectual*. La Habana : s.n., 2008.
- 35. Londoño, Alfonso Miranda.** *La Protección Jurídica de la Propiedad Intelectual en Cuba como modalidad del proteccionismo tecnológico*. Bogotá D.C : s.n., 2004.
- 36. Calatrava, María del Carmen.** *Sistema de administración de licencias y almacenamiento de creaciones intelectuales*. 2011.
- 37. Sunshine, Fabio Grobart.** *El Derecho de Propiedad Intelectual Internacional. Desafíos de Política para América Latina y el Caribe*. La Habana : s.n.
- 38. Fuentes, Indira Lilled Laurencio, Serafín Linares, Ana Mary y Miranda Pardo, Daily.** *El derecho de autor en la protección del software. El copyright y copyleft*. 2013.

ANEXOS

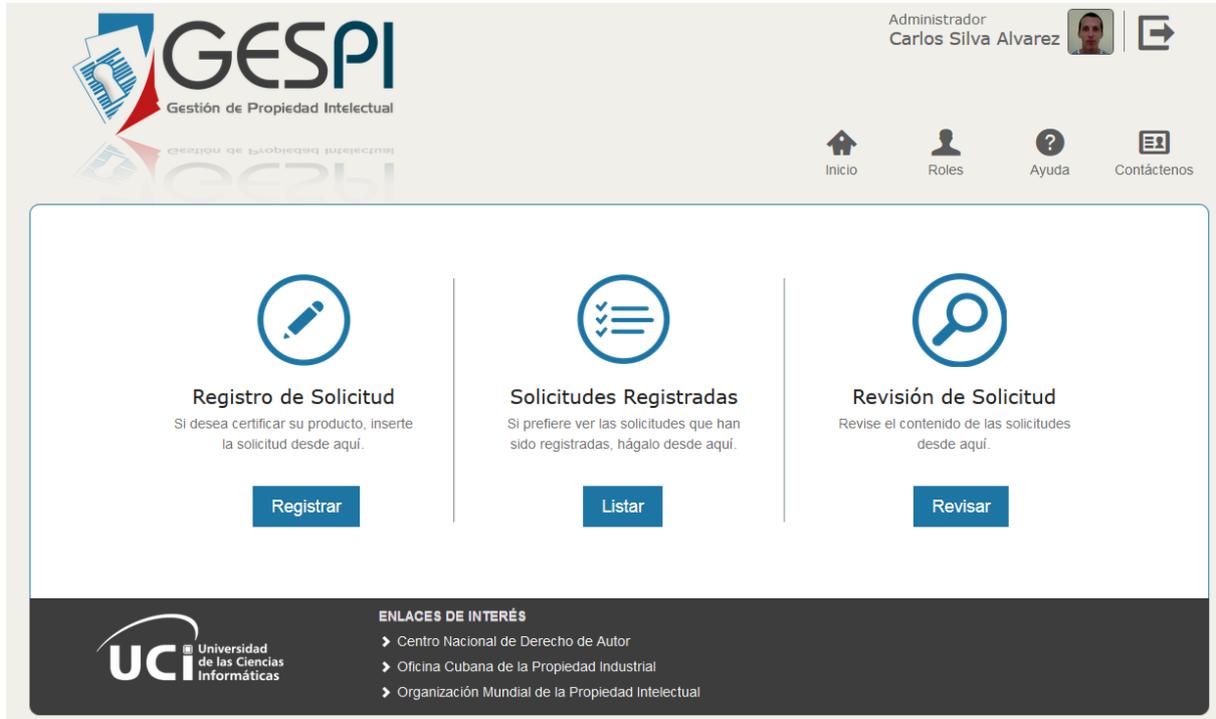


Figura 12: Prototipo de la página principal de la aplicación.

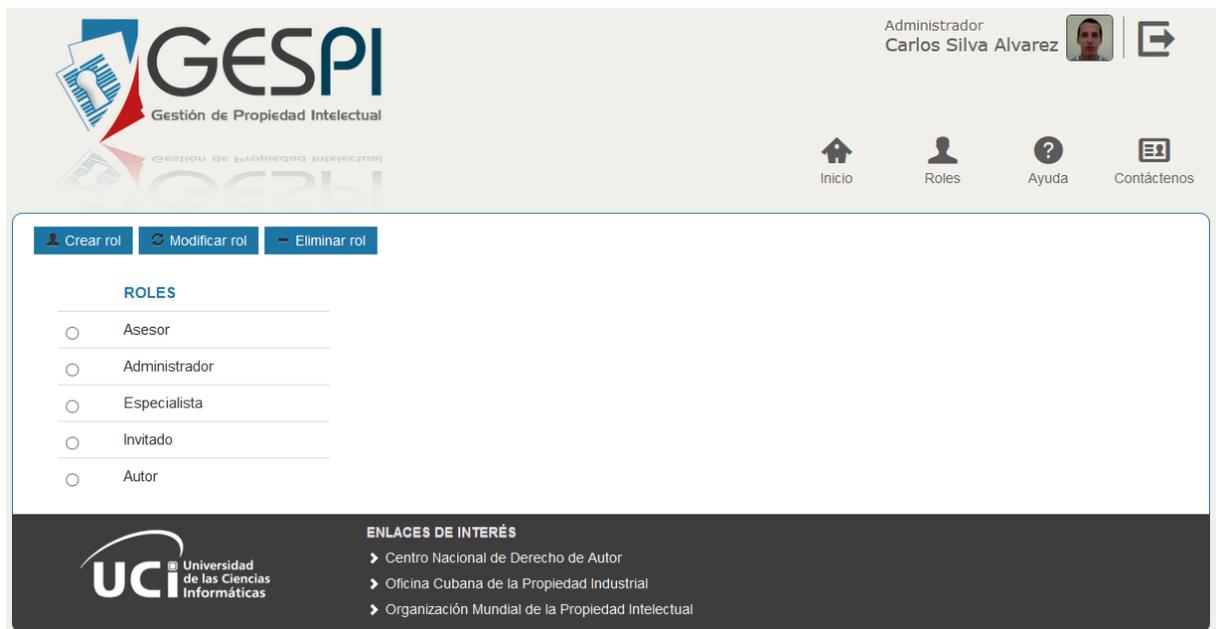


Figura 13: Prototipo de la página gestión de roles de la aplicación.

GESPI
Gestión de Propiedad Intelectual

Usuario

Contraseña

Iniciar sesión

Figura 14: Prototipo HU "Autenticar usuario".

Datos de solicitud de registro

Nota: Todos los campos son obligatorios

Nombre del Proyecto

Nombre del Producto

Versión

Nombre del Centro-Polo

- CEDIN
- CISED
- CENIA
- GEITEL

Cliente

Organismo

Datos del solicitante

Nombre y Apellidos

Usuario

Cargo

Documentación

Resumen

Figura 15: Prototipo HU "Insertar solicitud".

PRODUCTO	TIPO DE REGISTRO	USUARIO	FECHA	ESTADO				
SCADA-Almiqui	Software	betty	05-06-2014	Iniciado				
Sistema de Análisis Petrofísico	Software	yantunez	01-06-2014	Iniciado				
Sistema de Entrenamientos de Indicios Aduaneros	Software	rmferrer	28-04-2014	Iniciado				

Prev 1 Next

Figura 16: Prototipo de HU "Listar solicitud".

PRODUCTO	PROYECTO	TIPO DE REGISTRO	CENTRO	USUARIO	ESTADO				
SCADA-Almiqui	SCADA-Almiqui	Software	CEDIN	betty	Iniciado				
Sistema de Análisis Petrofísico	ANPER	Software	CEDIN	yantunez	Iniciado				
Sistema de Entrenamientos de Indicios Aduaneros	Entrenadores Aduaneros	Software	Centro de Informática Industrial	mferrer	Iniciado				

Prev 1 Next

Figura 17: Prototipo de HU "Revisar solicitud".

NOMBRE Y APELLIDOS	USUARIOS	CORREO ELECTRÓNICO	ROL		
Yaima Antunez Ojeda	yantunez	yantunez@uci.cu	Asesor		
Carlos Silva Alvarez	csilva	csilva@estudiantes.uci.cu	Administrador		

Figura 18: Prototipo de HU "Asignar rol".

Creación de Roles

Nombre del Rol

Crear rol

Figura 19: Prototipo de HU "Crear rol".