

Universidad de las Ciencias Informáticas  
Facultad 5



**“Sistema para la selección de modelos de negocio en la comercialización de los productos del Centro de Informática Industrial.”**

---

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

**Autor:**

Roisbel Portales López

**Tutores:**

MSc. Yaima Antúnez Ojeda

Ing. Aylín Rodríguez Pérez

Ing. Carlos Silva Alvarez

La Habana, Junio 2016

*“La vida no es fácil para ninguno de nosotros. ¿Pero qué hay con eso? Tenemos que tener perseverancia y, sobre todo, confianza en nosotros mismos.”*

*Marie Curie*



## **Dedicatoria**

A mi padre el cual estaría orgulloso de ver a su hijo hacerse ingeniero.

A mi mamá y a mí padrastro Albertico por apoyarme siempre en todo y hacer de mí la persona que hoy soy.

A mis hermanos y sobrino por ser lo más grande que tengo en este mundo y los quiero con la vida.

A mis abuelos Alejandro y Adelfa por haberme hecho el niño más feliz del mundo.

A mis abuelos Abilio y Caridad por preocuparse, por estar pendiente de mí.

A mis familiares y amigos.

## Agradecimientos

A la UCI y todo el universo que engloba porque, entre tragos amargos y dulces, me ha convertido en una mejor persona y me ha entregado momentos inolvidables.

A mi papá que aunque lo tuve a mi lado solo unos escasos 6 años de mi vida lo recuerdo con gran cariño. Espero que donde quiera que te encuentres estés orgulloso de cada cosa que he logrado en la vida. Se te extraña un montón.

A mi mamá por cada noche de desvelo, los buenos y los malos momentos que le he proporcionado, por darme fuerzas para seguir y no desfallecer, y confiar siempre en mí. Te quiero muchísimo y quiero que sepas que eres la mejor madre del mundo.

A mi padrastro Albertico al cual quiero y aprecio como a un padre, es que eso es lo que ha representado para mí en estos largos 17 años. Gracias por cada consejo, por cuidar de mí, de mi mamá y de mis hermanos por todo este tiempo y porque sin saberlo me impusiste el reto personal de hacerme ingeniero porque quería llegar a ser como tú, espero no defraudarte nunca.

A mis hermanos que los quiero con la vida y lo son todo para mí, a Alberto y Adriana les digo desde acá que siempre luchen por hacer realidad sus sueños y por hacerse personas de bien; a Yami muchas gracias por preocuparse tanto por mí y por darme ese maravilloso sobrino que es Daniel. A todos los quiero mucho.

A mis abuelos gracias por todo lo que hicieron por mí, los quise y quiero mucho.

A mi tía Rosalba por estar al tanto de mí en todo momento, por quererme tanto como a un hijo más, aunque a veces regañe por tardarme mucho en llamarla por teléfono, a mi tía Odalys porque aunque en la distancia siempre ha estado muy al tanto de mí y de mis estudios en todo momento y por quererme tanto como a un hijo más, las quiero muchísimo y espero que lo sepan.

A mis tíos Baby e Ifre por haberme dado todo su apoyo, cariño y haber estado al pendiente de mí durante todo este tiempo. Los quiero mucho.

A mis tíos Roberto, Miguel, Yaneisy, Salvador, Yadira y Ñico por preocuparse por mis estudios y por mí en general en todo este tiempo.

A mis primos Miguelito, Titico, Yanirys y Tonito por todo su cariño hacia mí.

A Jorge, Bertica, Jorge Alberto, Tía Olguita, Liset y Lianet por abrirme las puertas de su casa y de su corazón como un miembro más de la familia, saben que son muy especiales para mí.

A los buenos amigos que a lo largo de mi vida han perdurado y son parte de mi familia Mario, Reinaldo, Yudelkis, Roelkis, La China, Jose Carlos, Cepeda y Chichi gracias por su amistad incondicional.

A los amigos que me regaló la UCI (el mejor regalo de todos), sin dudas el mejor piquete de todos (El TEAM ROIS) Arletis (me demostró que siempre hay alguien más loco que tú y siempre estuvo presente su consejo cuando lo necesité), Rosmery (por ser tan buena amiga y por sus sabios consejos), Rodnier (por ser el hermano que siempre tuve en la UCI), Futiel y Ruben (gracias por su amistad incondicional) y Diomne (gracias por tu amistad y por aguantarme a tu lado en cada viaje a Ciego), en fin a todos gracias por aguantar mis pesadeces y locuras de vez en cuando y ayudarme a ser una mejor persona. Gracias por los momentos de alegría que hemos vivido juntos, por las noches de parchís y por cada travesura. Saben que son muy especiales para mí, gracias por existir, siempre los llevaré en mi corazón.

A Adrián por estar en mi lado en los momentos más difíciles y darme el ánimo necesario para en ocasiones no tirar la toalla. Por tener que soportarme que se no es fácil.

A otros amigos como son Arletis Ortiz, Claudia, Yaimara, Pedro, Yuneiry, Arianna, Yoan, Xiuny, Yoenia, Rosabel, Naidiley, Leonarkys, Eduardo, Alejandro Ravelo, Osniel, Eimé, Dayan, Aniel, Juan, Gómez, Ana Delia, en fin, a todos que son muchos.

A mis compañeros del apartamento 87 202 Pedro, Papo, Frank, Rigo, Jorge Luis, Rafa y Yugi.

A mis compañeros de la brigada 5501 (la mejor de todas) por todo este tiempo compartido y por cada experiencia vivida juntos.

A mis tutores por confiar en mí desde el primer momento, por exigir lo mejor de mí y guiarme con pasos certeros durante todo este tiempo. Les estaré eternamente agradecidos. Gracias.

A todos los profesores que de una forma u otra aportaron a mi formación como joven profesional, en especial Zaida porque además de profe en ocasiones la sentí mi madre.

En fin a todos los que de una forma u otra, permitieron que este trabajo fuera posible y quienes con mayor o menor esfuerzo, aportaron su granito de arena por ayudar.

## **Declaración de autoría**

Declaro que soy el único autor de este trabajo y autorizo al Centro de Informática Industrial de la Universidad de las Ciencias Informáticas; así como a dicha institución para que hagan el uso que estimen pertinente con este trabajo.

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

**Roisbel Portales López**

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

**MSc. Yaima Antúnez Ojeda**

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

**Ing. Carlos Silva Alvarez**

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

**Ing. Aylín Rodríguez Pérez**

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

## Resumen

Los modelos de negocio constituyen el arma fundamental de cada empresa, pues estos son los que permiten el desarrollo económico de las mismas. No existe consenso científico sobre la clasificación de todos los modelos de negocio existentes, pues estos varían según las particularidades de cada empresa y aún entre empresas de una misma rama suelen encontrarse diferencias en sus modelos de negocio. Por esta razón el Centro de Informática Industrial (CEDIN) de la Universidad de las Ciencias Informáticas (UCI) necesita una herramienta que automatice el proceso de selección de modelos de negocio para la comercialización de sus productos, pues actualmente este engorroso proceso se realiza de forma manual. Por tal motivo en la presente investigación se tiene como objetivo desarrollar un sistema informático para la selección de modelos de negocio en la comercialización de los productos del CEDIN. Surgiendo así el Sistema de selección de modelos de negocio (SESMO), el cual permite seleccionar de manera automática dichos modelos. El sistema fue desarrollado con el *framework* Symfony en su versión 2.8.3 y haciendo uso de un conjunto de tecnologías libres. Con la implantación del sistema se espera reducir el tiempo y esfuerzo empleado por los especialistas comerciales que conforman el Grupo de Inteligencia Empresarial (GIE) del CEDIN para la selección automática del modelo de negocio a emplear.

**Palabras clave:** comercialización, modelos de negocio, negocio, producto, *software*.

## Índice de Contenido

Introducción .....	10
Capítulo 1: Fundamentación teórica .....	13
Introducción .....	13
1.1 Modelos de negocios empleados en la comercialización de <i>software</i> .....	13
1.1.1 Modelos de negocio para el <i>software</i> privativo .....	14
1.1.2 Modelos de negocio para el <i>software</i> libre .....	15
1.1.3 Modelos de negocio empleados por el Centro de Informática Industrial en la comercialización de sus productos .....	18
1.2 Caracterización del proceso de selección de los modelos de negocio para la comercialización de los productos del CEDIN .....	19
1.3 Análisis de las posibles tecnologías, técnicas, herramientas y lenguajes a utilizar .....	20
1.3.1 Marco de Trabajo o <i>Framework</i> .....	20
1.3.2 Selección del framework para la manipulación de PHP .....	23
1.3.3 Lenguajes de programación .....	24
1.3.4 Lenguaje de modelado .....	26
1.3.5 Sistemas Gestores de Base de Datos .....	26
1.3.6 Selección de Sistema Gestor de Base de Datos .....	28
1.3.7 Servidor <i>web</i> .....	28
1.3.8 Herramientas a utilizar .....	29
1.3.9 Técnicas de Inteligencia Artificial empleadas para el proceso de toma de decisiones .....	31
1.3.10 Selección de la técnica de Inteligencia Artificial como apoyo en el proceso de toma de decisiones .....	32
1.3.11 Metodología de desarrollo de <i>software</i> .....	32
1.4 Conclusiones parciales .....	34
Capítulo 2: Análisis y diseño de la propuesta de solución .....	35
Introducción .....	35
2.1 Descripción de la propuesta de solución .....	35
2.1.1 Descripción del empleo del Sistema Basado en Reglas en la propuesta de solución .....	36
2.2 Actores del sistema .....	37

2.3 Requisitos del sistema.....	37
2.3.1 Requisitos funcionales .....	37
2.3.2 Requisitos no funcionales .....	40
2.4 Historias de usuarios .....	42
2.5 Arquitectura del sistema .....	47
2.6 Patrones de diseño del sistema.....	48
2.7 Diagrama de clases.....	51
2.8 Modelo de datos.....	51
2.9 Conclusiones parciales.....	52
Capítulo 3: Implementación y pruebas a la solución propuesta.....	53
Introducción .....	53
3.1 Implementación.....	53
3.1.1 Estándares de código .....	53
3.1.2 Diagrama de componentes .....	54
3.1.3 Diagrama de despliegue .....	55
3.2 Pruebas de <i>software</i> .....	57
3.2.1 Casos de prueba de Aceptación .....	58
3.2.2 Resultados de las pruebas .....	63
3.2.3 Comparación de la plataforma desarrollada con el proceso manual.....	64
3.3 Conclusiones Parciales .....	65
Conclusiones Generales.....	66
Recomendaciones .....	66
Referencias.....	67
Glosario de términos.....	70
Anexos.....	71
Historias de Usuario.....	71

## Índice de figuras

Figura 1 Flujo de trabajo empleado en la selección de modelos de negocio para la comercialización de los productos del CEDIN. Fuente: Elaboración propia. ....	19
Figura 2 Tabla de comparación realizada por la plataforma SESMO para el producto SCADA SAINUX. Fuente: Sistema SESMO.....	36
Figura 3 Patrón Arquitectónico Modelo-Vista-Controlador para Symfony 2. (29).....	48
Figura 4 Diagrama que representa el uso del Patrón Experto en el sistema. ....	50
Figura 5 Diagrama que representa el uso del Patrón Creador en el sistema. ....	50
Figura 6 Diagrama que representa el uso del Patrón Controlador en el sistema.....	51
Figura 7 Diagrama de clases de los requisitos Insertar, Modificar y Eliminar Modelos de Negocio. ....	51
Figura 9 Modelo de datos propuesto para el sistema.....	52
Figura 10 Diagrama de componentes de la aplicación.....	55
Figura 11 Diagrama de despliegue de la aplicación.....	56
Figura 12 Comportamiento de las no conformidades por iteración.....	63
Figura 13 Valoración de la plataforma SESMO teniendo en cuenta el tiempo.....	64

## Índice de tablas

Tabla 1 Fases de la metodología AUP-UCI. _____	33
Tabla 2 Actores del sistema. Fuente: Elaboración propia. _____	37
Tabla 3 Requisitos Funcionales del sistema. _____	38
Tabla 4 Requisitos no Funcionales del sistema. _____	40
Tabla 5 Historia de Usuario "Insertar Segmento de Mercado". _____	42
Tabla 6 Historia de Usuario "Modificar Segmento de Mercado". _____	43
Tabla 7 Historia de Usuario "Eliminar Segmento de Mercado". _____	44
Tabla 8 Historia de Usuario "Insertar Modelo de Negocio". _____	45
Tabla 9 Historia de Usuario "Modificar Modelo de Negocio". _____	46
Tabla 10 Historia de Usuario "Eliminar Modelo de Negocio". _____	46
Tabla 11 Caso de prueba de aceptación para la HU "Insertar Segmento de Mercado". _	58
Tabla 12 Caso de prueba de aceptación para la HU "Modificar Segmento de Mercado".	59
Tabla 13 Caso de prueba de aceptación para la HU "Eliminar Segmento de Mercado". _	60
Tabla 14 Caso de prueba de aceptación para la HU "Insertar Modelo de Negocio". ____	60
Tabla 15 Caso de prueba de aceptación para la HU "Modificar Modelo de Negocio". ____	61
Tabla 16 Caso de prueba de aceptación para la HU "Eliminar Modelo de Negocio". ____	62
Tabla 17 Historia de Usuario "Insertar Producto". _____	71
Tabla 18 Historia de Usuario "Modificar Producto". _____	72
Tabla 19 Historia de Usuario "Eliminar Producto". _____	74
Tabla 20 Historia de Usuario "Autenticar Usuario". _____	74
Tabla 21 Historia de Usuario "Insertar Usuario". _____	75
Tabla 22 Historia de Usuario "Modificar Usuario". _____	76
Tabla 23 Historia de Usuario "Eliminar Usuario". _____	77

## Introducción

En los inicios de la informática, el *software* no era considerado como un producto, sino un añadido a esta rama, incluido por los fabricantes de computadoras para que los clientes pudieran usar sus creaciones. Los programas se distribuían libremente entre programadores conjuntamente con su código fuente, con el fin de que estos pudieran adaptarlos según sus necesidades reales. Los sistemas de cómputo ganaron mayor aceptación en los usuarios; esto propició un aumento en la cantidad de computadoras y en el nivel de cómputo y sus prestaciones. Se hizo indispensable un crecimiento en la variedad y complejidad del *software* que comenzó a ser tan importante como las máquinas que operaban. Por lo que rápidamente se impusieron intereses comerciales que veían en esta nueva rama una industria lucrativa y prometedora.

A partir de ese momento fue indispensable establecer las pautas por las que se realizaría este proceso de comercialización en cada empresa productora de *software*, surgiendo así los modelos de negocio con este fin. Inicialmente los modelos de negocio estaban subvalorados debido al desconocimiento que se tenía de las ventajas que brinda su correcta utilización.

La dinámica de la economía actual hace que se produzcan cambios a una gran velocidad tanto en el área tecnológica como conceptual. Esto obliga continuamente a las empresas a innovar en nuevos modelos de negocio debido a la ventaja que implica encontrar el modelo adecuado y definirlo con claridad. Es importante conocer y fundamentar el modelo de negocio a utilizar, el cual debe de ser flexible permitiéndole a las empresas adaptarse rápidamente a los cambios del entorno que las rodea. Además, de que este constituye una guía que permite corregir el rumbo para lograr los objetivos propuestos y así concentrarse más en el mercado.

La Universidad de las Ciencias Informáticas (UCI) es una universidad con un modelo flexible de centro docente-productor que le permite formar profesionales altamente calificados y comprometidos con su país. Así como producir aplicaciones y servicios informáticos a partir de la vinculación estudio-trabajo, para servir de soporte a la industria cubana de la Informática. La UCI tiene dentro de su estructura a los centros productivos, encargados del desarrollo de productos y servicios especializados, resultados del proceso de investigación-desarrollo-innovación. Uno de ellos es el Centro de Informática Industrial (CEDIN) que tiene como misión: desarrollar productos y servicios informáticos de automatización industrial, con un alto valor agregado y que cumplan las necesidades y expectativas de los clientes, potenciando la formación especializada y la investigación.

En dicho centro existe el Grupo de Inteligencia Empresarial (GIE) el cual es el encargado de definir las estrategias de comercialización a utilizar con un producto o servicio, gestionar el portafolio de productos, los modelos de negocio, convenios y contratos del CEDIN. También realiza estudios sistemáticos de los productos y empresas de la competencia, propone precio a los productos y servicios del centro y se encarga de dirigir la elaboración de ofertas de productos y servicios del mismo. Gestiona el proceso de registro de productos y servicios del Centro, así como los procedimientos económicos detrás de cada proyecto en función de lo definido en las ofertas y lo firmado en los contratos con los clientes.

Para la selección de los modelos de negocio más adecuados el GIE debe estudiar los mercados objetivos, realizar un estudio de la competencia en el mercado, estudiar los modelos de negocio empleados en productos similares al que se desea comercializar. Se debe además, definir los costos para cada actividad a realizar para dicha comercialización, establecer la estrategia de precio para el producto, describir el modelo de negocio a utilizar y aprobar el modelo de negocio escogido.

En la actualidad el proceso de selección de modelos de negocio se realiza de forma manual lo cual se torna engorroso, ya que incrementa el tiempo y esfuerzo por parte del personal que conforma el GIE. Además, se pueden cometer errores durante el procedimiento. Como fue explicado anteriormente, este proceso involucra numerosas tareas, las que generan su respectiva documentación; dicha documentación es almacenada únicamente en las computadoras de los especialistas a cargo del proceso, lo que provoca que no exista completa seguridad en estos archivos, que puedan extraviarse o corromperse y que el acceso a dicha información no esté centralizado. Además, el proceso se puede ver afectado ya que la aprobación de la selección se realiza en encuentros personales con el Jefe de Centro (reuniones, encuentros, entre otros), retrasando así la entrega del producto al cliente.

Teniendo en cuenta la situación problemática antes expuesta se plantea como **problema a resolver**: ¿Cómo agilizar el proceso de selección de los modelos de negocio para la comercialización de los productos del CEDIN?

Con el fin de dar solución al problema planteado se define como **objetivo general**: Desarrollar un sistema informático para la selección de modelos de negocio en la comercialización de los productos del CEDIN.

En la presente investigación el **objeto de estudio** lo constituyen: los modelos de negocio para la comercialización de *software*.

Por todo lo anterior se precisa como **campo de acción**: los sistemas informáticos de selección de modelos de negocio para la comercialización de *software* industrial.

Para darle cumplimiento al objetivo general se trazaron las siguientes **tareas de investigación**:

- ✓ Elaboración del marco teórico de la Investigación a partir del estado del arte existente sobre el tema.
- ✓ Caracterización del procedimiento de selección de los modelos de negocio para la comercialización de los productos del CEDIN.
- ✓ Selección de una técnica de Inteligencia Artificial que realice el proceso de toma de decisiones, para su futura utilización en el desarrollo de la propuesta de solución.
- ✓ Valoración de las soluciones existentes que responden al problema de la investigación, sus limitaciones y fortalezas.
- ✓ Selección de las principales herramientas, tecnologías, lenguajes y metodologías a utilizar para la construcción de la propuesta de solución.
- ✓ Realización del análisis de la propuesta de solución.
- ✓ Realización del diseño de la solución propuesta.
- ✓ Implementación de la solución propuesta.
- ✓ Desarrollo del proceso de pruebas y validación del sistema implementado.

Para la realización de la investigación fueron aplicados los siguientes **métodos de investigación científica**:

**Métodos teóricos:**

Analítico-Sintético se utilizó en la búsqueda y análisis de los elementos más importantes asociados a los modelos de negocio empleados para la comercialización de *software* según la bibliografía consultada en libros, páginas *web* y artículos, realizando una síntesis de los elementos más importantes de la misma.

Histórico-Lógico se empleó para lograr una mayor comprensión del estado y tendencias actuales de los modelos de negocio empleados para la comercialización de *software*, así como las tecnologías más usadas para el desarrollo de plataformas *web*.

Modelación se utilizó para la modelación de diagramas de componentes, de paquetes, entre otros, para la implementación de la propuesta de solución.

**Métodos empíricos:**

La Observación se tuvo en cuenta a la hora de valorar la propuesta más adecuada de las tecnologías y soluciones existentes, además de permitir la observación de diferentes sitios *web* para conocer su funcionamiento, organización y estructura de la información que contiene.

La presente investigación está estructurada en 3 capítulos de la siguiente manera:

**Capítulo 1: “Fundamentación teórica”:** en este capítulo se abarcará el estado de los sistemas de selección de modelos de negocio a nivel nacional e internacional, para posteriormente realizar un balance y estudio de un conjunto de herramientas, tecnologías y metodología a seleccionar, para el desarrollo de la propuesta de solución.

**Capítulo 2. “Análisis y diseño de la propuesta de solución”:** en este capítulo se expone la propuesta y características del sistema de selección de los modelos de negocio de manera detallada, describiéndose los requisitos funcionales y no funcionales, historias de usuario, diagrama de clase, modelo de datos, así como todo lo referente a los patrones utilizados.

**Capítulo 3: “Implementación y pruebas a la solución propuesta”:** en este capítulo se detalla la construcción y validación de la solución al problema planteado. Además, se realizan y describen las pruebas de *software* a la aplicación, en este caso se utilizarán las pruebas funcionales, más específicamente el nivel de pruebas de aceptación.

## **Capítulo 1: Fundamentación teórica**

### **Introducción**

En el presente capítulo se precisan un conjunto de elementos teóricos que van a conllevar a un mejor entendimiento de la presente investigación. Adicionalmente se realiza un estudio del estado del arte de los modelos de negocio existentes para la comercialización de *software*. Se exponen además, la selección de las técnicas, herramientas y metodologías que se emplearán para el desarrollo de la propuesta de solución.

#### **1.1 Modelos de negocios empleados en la comercialización de *software***

“El modelo de negocio o *bussines model* del inglés, es un modelo o diagrama de cómo deberá funcionar un negocio para ofrecer valor a su mercado objetivo y producir ganancias e indica claramente cómo generar las mismas mediante un correcto posicionamiento en la cadena de valor.” (1)

Usualmente un modelo de negocio se presenta como un resumen planificado que abarca tanto el concepto de estrategia como el de implementación y debe responder a las siguientes preguntas: (1)

1. ¿Cuál es mi público objetivo o mis clientes potenciales?
2. ¿Cómo diferenciamos nuestra oferta de productos y/o servicios de los de la competencia?
3. ¿Cómo añadimos valor a nuestra oferta para que esta sea percibido por nuestros clientes?
4. ¿Cómo obtenemos y fidelizamos a nuestros clientes?
5. ¿Cómo nos damos a conocer en el mercado? (estrategia publicitaria).
6. ¿Cuáles son las actividades a desarrollar y como están planificadas?
7. ¿Cómo utilizamos de manera óptima nuestros recursos humanos, materiales y financieros?

Es bueno conocer que no existe una clasificación clara para todos los modelos de negocio existentes, pues estos varían según las particularidades de cada empresa y aún entre empresas de una misma rama suelen encontrarse diferencias en sus modelos de negocio. También la innovación acelerada en los mismos da lugar a nuevos modelos y a cambios continuos.

Como bien es conocido en la industria del *software* existen dos tipos de productos en términos comerciales, los de código privativo y los de código abierto o libre. Pero cada empresa teniendo en cuenta el tipo de código de sus productos utiliza varios tipos de modelos de negocio. A continuación, se describen algunos de los modelos de negocio más utilizados por ambas industrias a la hora de comercializar sus productos:

### **1.1.1 Modelos de negocio para el *software* privativo**

En el caso del *software* privativo no se puede analizar sin tener en cuenta el dominio monopolista del mercado que ejercen la mayoría de las grandes empresas del sector que basan su modelo en la venta de licencias de uso y control del conocimiento, logrando así que sus aplicaciones dominen el mercado. Para ello, cualquier estrategia que impida o limite el uso de la tecnología de la competencia, es válida. Esto trae consigo que el conocimiento y la tecnología estén al servicio de intereses empresariales concretos y no de la sociedad.

#### **Ejemplos:**

### **Venta de licencias de uso**

El principal modelo de negocio del software privativo se basa en la explotación de la propiedad intelectual del *software*, con la sesión del derecho de uso a través de la venta de una copia ya sea en CD, DVD, a través de Internet, entre otros. Pero limitando los derechos que “debilitan” la propiedad intelectual como estudio, adaptación, distribución y mejora a través de la implementación de leyes de *copyright* y la reserva del código fuente solo para uso de la empresa propietaria. Este modelo permite la definición de estrategias variadas en función de una variedad enorme de modos de explotación y licenciamiento. Ejemplos de compañías del sector que aplican este modelo: Microsoft, Oracle, Adobe. (2)

### **Venta de servicios asociados**

El mercado de *software*, tanto libre como propietario, se ha venido enfocando, desde hace años a la venta de servicios por lo que otra fuente de ingresos importantes del *software* propietario se centra, aunque no son los únicos, en los siguientes servicios: consultoría, soporte, formación y desarrollos a la medida los cuales están configurados como “paquetes” o grupo de servicios según la conveniencia de la empresa comercializadora a lo cual hay que añadir también la venta de actualizaciones periódicas. Las empresas productoras de *software* privativo generalmente utilizan combinaciones de venta de licencias de uso y servicios que pueden ser añadidos a la compra inicial del producto o contratados posteriormente aunque hay muchas empresas que basan su modelo de negocio en los servicios asociados a un determinado producto. (2)

#### **1.1.2 Modelos de negocio para el *software* libre**

En el caso del *software* libre las libertades que este confiere a sus usuarios y desarrolladores, hace casi inviable el sistema de ventas de licencias de uso utilizado en el *software* propietario, aunque no es totalmente imposible. Esto hace que las empresas que centran su negocio entorno al *software* libre basen sus estrategias en la venta de servicios asociados a un determinado producto o línea de productos que generalmente se ofrece gratis y por tanto los modelos de negocio relacionados a continuación se basan en este principio. Estos modelos de negocio son los más usuales en la industria, pero es imposible abarcar la diversidad de modelos de negocio que hoy existen fruto de la innovación constante a la que se ven obligadas las empresas del sector. De manera casi similar, aunque con una óptica diferente los servicios más comunes asociados a un producto de

*software* tanto libre como propietario son: Consultoría, Formación, Personalización o Adaptación, Desarrollos a la medida, Integración, Implantación y Soporte.

**Ejemplos:**

**Vendedor de Soporte o *Support Seller***

Este modelo es uno de los más atractivos y prometedores para las empresas del sector ya que a medida que aumente la penetración del *software* libre la demanda de este servicio aumentará considerablemente. Siguiendo este modelo, una empresa promueve un producto libre generalmente desarrollado por ella o en cuyo desarrollo participa activamente y una vez creado un mercado para este, la empresa comienza a vender los servicios generados entorno al producto como son: consultoría, personalizaciones o adaptaciones, soporte, entre otros. (2)

Dado el carácter abierto de la tecnología, aquellas empresas de *software* libre que no tienen tecnología creada y desarrollada por ellos mismos, basan el soporte en el conocimiento propio de la aplicación implantada (libre), además de los canales de información que se han creado en las comunidades.

Por su parte las empresas que sí desarrollan tecnología pueden ofrecer el soporte directamente debido al conocimiento que tienen de las mismas. Los servicios asociados al soporte, se ofrecen de diferentes formas, flexibilizando al máximo las condiciones, ya que la libertad de la tecnología te lo permite, tal y como hemos visto anteriormente.

***Widget Frosting***

Este modelo se centra fundamentalmente en la venta de hardware, y el *software* libre, instalado en el mismo se considera un complemento para él, que puede ayudar a la empresa a obtener una ventaja competitiva ya sea desde el punto de vista de la variable precio, el cual puede fijar por debajo del de la competencia con el lógico aumento en sus niveles de ventas o manteniendo sus precios para obtener un margen de ganancia superior. Esto se logra por la reducción de costos que implica prescindir de las licencias de uso de *software* privativo.

Aun cuando con el uso de este modelo los fabricantes de hardware se benefician del *software* libre algunos autores no lo consideran propiamente como un modelo de negocio de *software* libre. (2)

### **Consultoría**

Para poder ofrecer soluciones informáticas a las empresas, se necesita de análisis organizacionales de las mismas para detectar cuáles son sus necesidades. Esta línea de negocio consiste en brindar un servicio de análisis de la situación de la empresa para recomendar soluciones que satisfagan las necesidades encontradas. A diferencia de las consultorías de *software* propietario, aquí se tiene una gama enorme de posibilidades tecnológicas. La gama de posibilidades estará en muchas ocasiones directamente relacionada al conocimiento tecnológico que se tenga. La empresa de *software* propietario sólo puede ofertar su gama de productos, ya que ahí es donde tiene su negocio. La empresa de *software* libre por su parte, ofrece la solución libre que más se adapta a las necesidades de la empresa cliente, sin entrar en otras consideraciones. (2)

### **Desarrollos a la medida**

En muchas ocasiones, a raíz de la consultoría realizada, la empresa se embarca en algún desarrollo específico. Bases de datos, *websites*, aplicaciones para un sector, entre otros. Son casos en los que las empresas de *software* libre ofrecen sus servicios de desarrollo a las empresas adaptando o creando *software* según las necesidades de la misma. Esto se contrapone al modelo propietario donde en la gran mayoría de los casos es la empresa quien tiene que adaptarse al *software*. (2)

### **Integración/Implantación**

La implantación/integración es otra de las líneas de negocio que las empresas llevan a cabo. La implantación de la tecnología y la integración de la misma con estructuras existentes en la actualidad, son otros de los servicios que las empresas de *software* libre ofrecen en la implantación de su tecnología. (2)

### **Capacitación o Formación**

Dado el carácter libre de la aplicación, y de la extensa y libre documentación existente en la red, cualquier empresa que cuente con los conocimientos técnicos y docentes necesarios, puede ofrecer los servicios de formación. Frente a esto, las empresas de *software* propietario ofrecen servicios de formación normalmente a través del canal comercial. Debido a la existencia de intermediarios y del contenido privado de la aplicación, los costes asociados a la formación de *software* propietario serán siempre a medio/largo plazo más costosos que los de *software* libre. (2)

### **Franquicia del *software***

El modelo de negocio teórico consiste en que, a medio plazo, aquella empresa de *software* libre que logre un reconocimiento en el mercado, podrá franquiciar cada una de las líneas de negocio que tenga a terceros para que las usen bajo su marca. Esta forma de generar negocio sería una salida frente a un crecimiento interno de la empresa para alcanzar un mercado en particular. (2)

### **Liberación de aplicación**

Está definido como el paso de una aplicación propietaria a una libre directamente. Este modelo de negocio tiene unas características muy definidas ya que la empresa se convierte en centro de consultoría e implantación, centro de formación y soporte tanto para las grandes empresas como para la comunidad. La empresa al fin y al cabo ofrecerá el conocimiento implícito. (2)

### **Licencia dual**

Este modelo de negocio es complementario a algunos de los que acabamos de ver. La existencia de licencias duales en tecnología libre permite a las empresas mantener un producto libre y gratis y por otra parte ofertar una versión o complementos comerciales. Una de las empresas que venden bajo doble licencia es MySQL. MySQL AB tiene sus líneas de negocio basadas en tres diferentes áreas: soporte comercial y servicios de suscripción a los usuarios de MySQL.com, venta de licencias a terceros, y franquicia de la marca. Concretamente MySQL AB ofrece su aplicación de base de datos bajo dos licencias diferenciadas, una bajo GPL<sup>1</sup> y otra bajo licencia de uso. (2)

#### **1.1.3 Modelos de negocio empleados por el Centro de Informática Industrial en la comercialización de sus productos**

En los años de experiencia del CEDIN en el desarrollo de *software*, el Grupo de Inteligencia Empresarial ha logrado comercializar varios productos, bajo los modelos de negocio Venta de licencia de uso y Venta de servicios asociados a un producto; ya que estos eran los más adecuados para su comercialización. Esto no significa que otros modelos de negocio no

---

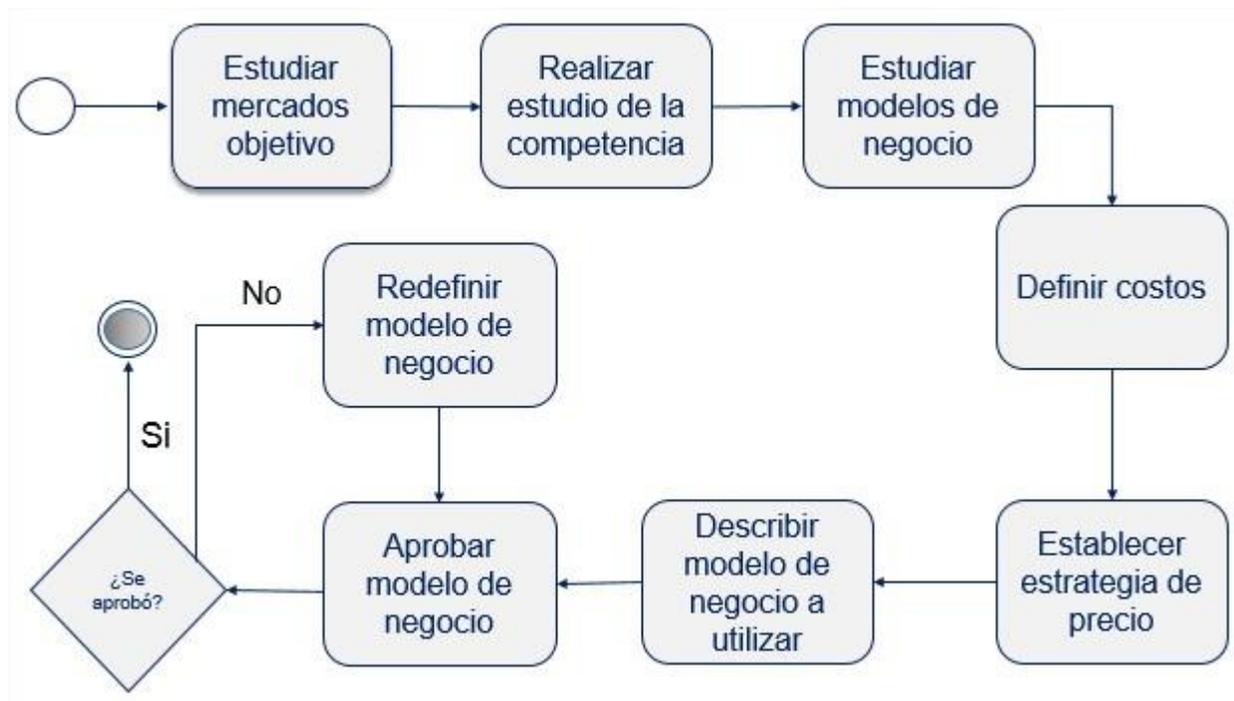
<sup>1</sup> **GPL**: Licencia Pública General o en inglés *General Public License*.

puedan ser elegidos por los especialistas comerciales, para realizar el proceso de comercialización.

## 1.2 Caracterización del proceso de selección de los modelos de negocio para la comercialización de los productos del CEDIN

Actualmente el proceso de selección de los modelos de negocio para la comercialización de los productos del CEDIN se realiza de manera engorrosa, ya que el GIE de dicho centro debe de ejecutar una serie de procesos para lograr la selección más óptima y la más adecuada.

A continuación, se muestra el flujo de trabajo del proceso de selección del modelo de negocio a emplear en la comercialización de un producto del CEDIN.



**Figura 1** Flujo de trabajo empleado en la selección de modelos de negocio para la comercialización de los productos del CEDIN. Fuente: Elaboración propia.

Este proceso de selección comienza cada vez que el centro cuenta con un producto para ofertar en el mercado. Seguidamente el GIE debe realizar un estudio del sector de mercado al cual está dirigido dicho producto. Por cada sector de mercado seleccionado se debe elaborar un documento que refleje las principales características del mismo. Luego se realiza el estudio de los modelos de negocio aplicados a productos similares por parte de

la competencia, con el objetivo de seleccionar el tipo de modelo de negocio a tener en cuenta para el producto. Después se realiza una comparación entre estos productos y el producto a comercializar del centro, con el objetivo de determinar la competitividad de dicho producto con respecto a los de la competencia. Este proceso se puede tomar un tiempo algo largo, ya que la mayoría de las empresas no hacen públicos los modelos de negocio empleados por ellas para la comercialización de sus creaciones.

También el GIE debe establecer la estrategia de precio para el producto, a partir de la ficha de costos por la que se rige la universidad, teniendo en cuenta el costo y esfuerzo que estos generan. Otra de las tareas realizadas para la selección del modelo de negocio a utilizar es el establecimiento de los costos de cada actividad básica a realizar para la comercialización del producto. Para ello se elabora un nuevo documento el cual respaldará el costo y precio final del producto, y en el caso de los proyectos por los que no se perciba ingresos, se debe calcular el costo, aunque no se ponga un precio.

Seguidamente se selecciona y describe el modelo de negocio a utilizar para la comercialización del producto, definiendo así dicho modelo. Después esta propuesta es revisada y aprobada por el director del CEDIN. Este modelo también puede ser revisado por otros miembros del consejo de dirección del centro, los cuales pueden aportar los elementos técnicos que consideren necesarios al mismo. Una vez que el modelo de negocio ha sido revisado y firmado por los involucrados, el mismo se archiva para la comercialización del producto. De esta forma se le da fin al proceso de selección de los modelos de negocio a utilizar en la comercialización de un producto del centro.

### **1.3 Análisis de las posibles tecnologías, técnicas, herramientas y lenguajes a utilizar**

En este epígrafe se evaluarán y seleccionarán las tecnologías, técnicas y herramientas más idóneas para el desarrollo de la solución, al igual que la metodología de desarrollo a seguir para el logro del producto deseado.

#### **1.3.1 Marco de Trabajo o *Framework***

Un *framework web*, está diseñado para apoyar el desarrollo de sitios *web* dinámicos. Ofrece un conjunto de componentes para acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. (3)

Se desprende de lo anterior la necesidad de utilizar estos en la presente solución, permitiendo una estructura conceptual y tecnológica de soporte para el desarrollo de sus funcionalidades.

### **Bootstrap**

Es el *framework* de twitter que permite crear interfaces *web* con CSS y Javascript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador o de una *Tablet* sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo o *Responsive Design* (4). Entre sus características que hicieron posible su selección, se destaca interfaces de gran usabilidad, *Responsive Design* de serie e integración de la librería jQuery para diferentes efectos.

Atendiendo al autor Cuervo.V resalta características distintivas en Bootstrap como son (5):

- Desarrollo de aplicaciones *Responsive Design*.
- Soporte HTML5 y CSS3.
- Permite desarrollo de aplicaciones para escritorio y móvil.
- Soporte multi-navegador, incluido Internet Explorer.
- Permite definir diferentes diseños.
- Amplio conjunto de componentes para el desarrollo.
- Posibilidad de compilar el *framework* con diferentes valores.

### **Symfony**

Es un *framework* diseñado para optimizar el desarrollo de aplicaciones *web* gracias a sus diversas características. Separa las reglas de negocio de la aplicación, la lógica del servidor y las vistas de presentación, lo cual constituye un aspecto importante a la hora de desarrollar cualquier solución. Cuenta con muchas herramientas y clases orientadas a disminuir el tiempo de desarrollo de una aplicación *web* compleja. Además, automatiza las tareas más comunes para que el desarrollador pueda ocuparse completamente de los aspectos específicos de cada aplicación. Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales. Es compatible con la mayoría de

gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft.  
(6)

A continuación, se hace mención de las principales características del *framework*:

- Gran facilidad de instalación y configuración en la mayoría de las plataformas de Linux y Windows.
- Independiente del manejador de base de datos: utiliza Propel o Doctrine, una capa de abstracción que le permite interactuar con varias bases de datos.
- Sencillo de usar en la mayoría de los casos y lo suficientemente flexible para adaptarse a escenarios complejos.
- Se rige por las mejores prácticas y patrones de diseño para la *web*.
- Preparado para aplicaciones empresariales y adaptables a las políticas de arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Manejo de memoria caché: lo cual reduce el uso del ancho de banda y la carga en servidor.
- Contiene código fácil de leer, el cual incluye comentarios de phpDocumentor y formas de adaptarlo en forma sencilla.
- Mecanismos de autenticación y credenciales: que facilitan la creación de secciones restringidas y la gestión de seguridad de usuarios.
- URLs<sup>2</sup> inteligentes: que permiten que las direcciones de las páginas *web* sean parte de la interfaz y resulten amigables a los motores de búsqueda.
- Fácil de extender, lo que posibilita integrarse con librerías desarrolladas por terceros.
- Incorpora herramientas que facilitan la prueba y depuración de aplicaciones: como unidades de generación de código, pruebas del funcionamiento del *framework*, panel de depuración, interfaz por línea de comandos y configuración en tiempo real.

## Yii

Yii es un *framework* para PHP de alto rendimiento basado en componentes *web* para desarrollar aplicaciones de gran escala. Permite una máxima reusabilidad en la

---

<sup>2</sup> **URLs**: Localizador Uniforme de Recursos o del inglés *Uniform Resource Locator* en plural.

programación *web* y puede acelerar significativamente el proceso de desarrollo. El nombre Yii proviene de facilidad, eficiencia y extensión.

Es un *framework* genérico para programar *web* que puede ser utilizado para desarrollar virtualmente cualquier tipo de aplicaciones *web*. Pues es liviano y está equipado con las soluciones más sofisticadas, está especialmente diseñado para trabajar con aplicaciones *web* de tráfico alto, como portales, foros, sistemas de administración de contenidos (CMS), sistemas de comercio electrónico (*e-commerce*), y otros. (7)

Este implementa el patrón de diseño Modelo Vista Controlador (*Model-View-Controller* (MVC)) el cual es adoptado ampliamente en la programación *web*. MVC tiene por objeto separar la lógica del negocio de las consideraciones de la interfaz de usuario para que los desarrolladores puedan modificar cada parte más fácilmente sin afectar a la otra. En MVC, el modelo representa la información (los datos) y las reglas del negocio; la vista contiene elementos de la interfaz de usuario como textos, formularios de entrada; y el controlador administra la comunicación entre la vista y el modelo. Más allá del MVC, Yii también introduce un *front-controller* llamado aplicación, el cual representa el contexto de ejecución del procesamiento del pedido. La aplicación resuelve el pedido del usuario y las despacha al controlador apropiado para tratamiento futuro.

Las principales características del *framework* Yii se mencionan a continuación:

- Fácil de instalar.
- Puede correr en modo *debug* o en modo producción, acorde al valor de la constante YII\_DEBUG, por defecto toma modo producción.
- Liviano de correr y equipado con soluciones de cacheo sofisticadas.
- Permite máxima reutilización.
- Ha sido probado en Apache HTTP server en Windows y en Linux. Pero también deberá correr en otros servidores *web* o plataformas que soporten PHP 5.

### 1.3.2 Selección del framework para la manipulación de PHP

Se elige como marco de trabajo del lado del servidor a Symfony 2 por ser multiplataforma y destacar sobre los otros *frameworks* PHP, por su eficiencia y su rica librería de funcionalidades, así como también su clara documentación. Además, está diseñado para optimizar el desarrollo de las aplicaciones *web*, ya que separa la lógica del negocio, la lógica de servidor y la presentación de las mismas. Este *framework* está desarrollado

completamente en el lenguaje PHP 5. También proporciona herramientas y clases encaminadas a reducir el tiempo de desarrollo.

### 1.3.3 Lenguajes de programación

Partiendo de la selección del *framework* Symfony para el desarrollo de la aplicación *web* se hace necesario hacer una selección de los lenguajes de programación y de modelado para realizar el análisis y diseño, caracterizándose los mismos en el presente acápite.

Teniendo en cuenta esto se hace la elección del lenguaje de programación PHP 5 para el desarrollo e implementación de funcionalidades del sistema. La utilización de JavaScript 1.8, HTML5 (Lenguaje Extensible de Marcado de Hipertexto) y CSS3 (Hojas de estilo en cascada o *Cascading Style Sheets*) permitirán complementar esta selección de desarrollo debido a sus potencialidades a la hora de realizar un diseño *web* y de brindar una interfaz intuitiva y amigable al usuario en conjunto con el *framework* de Twitter Bootstrap.

#### HTML5

Lenguaje de Marcado de Hipertexto o HTML. Lenguaje compuesto por cientos de etiquetas o marcas que permite definir el contenido y la apariencia de una página *web* en gran medida, además puede incluir script como JavaScript el cual afecta el comportamiento de los navegadores y otros procesadores de HTML, define una estructura de documento jerárquica, con elementos y componentes interconectados. (8)

#### PHP5

Es un lenguaje de programación de código abierto muy popular especialmente adecuado para el desarrollo *web*. Permite la creación de aplicaciones *web* con interfaz gráfica, se conecta a servidores de bases de datos tales como Oracle, MySQL y PostgreSQL y es multiplataforma<sup>3</sup> (9). Debido a su amplia distribución, PHP está perfectamente soportado por una gran comunidad de desarrolladores, caracterizándose por ser un lenguaje muy rápido y de alto nivel, totalmente compatible con los modernos métodos orientados a objetos, prácticas y principios.

---

<sup>3</sup> Sistemas operativos: Unix, Windows, Linux y Mac OS.

## CSS3

CSS desarrollada por la W3C (*World Wide Web Consortium*), es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML que separa los contenidos y su presentación por lo que es imprescindible para crear páginas *web* complejas (10). Permite definir la apariencia de cada elemento de la aplicación *web*: colores, fondos, márgenes, bordes, tipos de letra, modificando la apariencia y posición de cada elemento dentro de la página. Lo que permite controlar el estilo y formato de sus documentos.

Ofrece, además:

- Fácil creación de las plantillas al mantener la misma imagen en todas las páginas del sitio.
- Mayor accesibilidad y limpieza del código fuente.
- Código HTML más legible.
- Logra que los documentos se vean igual en todos los navegadores.
- Optimización de los tiempos de carga y el tráfico del servidor.

## JavaScript

Los siguientes autores Mora (8) y Zakas (11) constatan que es un lenguaje interpretado que no necesita ser compilado, solo es necesario un navegador para su interpretación por lo que se ejecuta del lado del cliente. Creado con el objetivo de hacer páginas *web* dinámicas, es orientado a objeto y guiado por eventos que produce el mismo usuario, haciendo amena la interactividad con las páginas mediante efectos tales como: cambio de color de algunos elementos de la página, crear páginas interactivas con programas como calculadoras, agendas, tablas y calendarios.

Es un lenguaje multiplataforma siendo interpretado por la mayoría de los navegadores modernos. Sencillo, rápido y fácil de aprender por personas de poca experiencia. Permite reducir la carga en el servidor, ya que los datos incorrectos se filtran en el cliente y no se envían al servidor.

## jQuery

jQuery es una biblioteca de JavaScript rápida y concisa que simplifica el trabajo con documentos HTML y XHTML, manejo de eventos, animación y las interacciones Ajax<sup>4</sup> para el desarrollo *web* rápido. Está diseñado para cambiar la forma en que escribes JavaScript. Es multiplataforma y para comenzar a utilizarlo simplemente se debe incluir un script que contiene el código de jQuery. (12)

### 1.3.4 Lenguaje de modelado

#### UML

El Lenguaje de Modelado Unificado o UML por sus siglas en inglés (*Unified Modeling Language*) es un lenguaje estándar y gráfico para la modelación de sistemas de *software*. Está respaldado por la OMG (*Object Management Group*) y permite describir métodos o procesos, visualizar, especificar construir y documentar un sistema (13). UML es gratuito, accesible a todos, y conforma la colección de las mejores técnicas de ingeniería en el desarrollo de *software*. Por ser un lenguaje estándar y reconocido en el desarrollo de *software* profesional a nivel internacional se utiliza UML.

### 1.3.5 Sistemas Gestores de Base de Datos

Los sistemas de gestión de base de datos o en inglés *Database management system* (DBMS), según Álvarez (14) se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos, ayudando a realizar las siguientes acciones:

- Definición de los datos
- Mantenimiento de la integridad de los datos dentro de la base de datos
- Control de la seguridad y privacidad de los datos
- Manipulación de los datos

Algunos ejemplos de DBMS son Oracle, DB2, PostgreSQL, MySQL, y MS SQL Server, siendo entre ellos los más reconocidos y populares dentro del *software* libre MySQL y PostgreSQL. Estos dos potentes gestores de base datos se integran al Symfony haciendo de esta una excelente herramienta para el desarrollo *web*.

---

<sup>4</sup> **Ajax**: JavaScript asíncrono y XML o del inglés *Asynchronous JavaScript and XML*.

## MySQL

MySQL es un sistema gestor de bases de datos muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo. (15)

Está disponible para múltiples plataformas tanto libre como privativas, es multi-hilo y multi-usuario, pudiéndose utilizar como cliente-servidor o incrustado en aplicaciones. Facilita la integración con programas desarrollados en C y C++ pues está desarrollado en este lenguaje.

## PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (*Berkeley Software Distribution* o en su concepto en español Distribución de *software* Berkeley) y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multi-hilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. (16)

Es compatible con el almacenamiento de objetos binarios de gran tamaño, como imágenes, sonidos o vídeo. Cuenta con interfaces de programación nativas para C / C ++, Java, NET, Perl, Python, Ruby, Tcl, ODBC, entre otros lenguajes de programación. Cuenta con características sofisticadas como la replicación asincrónica y sincrónica, transacciones anidadas (puntos de retornos), “*tablespaces*”, copias de seguridad en caliente, múltiples métodos de autenticación, MVCC (Control de Concurrencia Multi-Versión) el cual permite que se ejecuten sobre una tabla varias transacciones a la vez y un sofisticado planificador consulta / optimizador. (21)

Se hace notable entre sus características su buen funcionamiento con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Es un sistema

multiplataforma disponible para Linux y Windows en todas sus variantes, presentando una extensa y gran variedad de documentación.

### **1.3.6 Selección de Sistema Gestor de Base de Datos**

Después de haber realizado un análisis de dos de los SGBD más utilizados en el mundo dada sus características y prestaciones, se aprecian las ventajas que brinda PostgreSQL. Dado que este sistema provee de gran capacidad de almacenamiento, consistencia, escalabilidad y rendimiento bajo grandes cargas de trabajo. Además, es distribuido por la Licencia de *Software Libre Permisiva* (BSD) y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado. Utiliza un modelo cliente/servidor y usa multiprocesos garantiza la estabilidad del sistema. Por estas razones se decide desarrollar el sistema bajo el SGBD PostgreSQL en su versión 9.3.

### **1.3.7 Servidor web**

Los servidores *web* son aquellos cuya tarea es alojar sitios y/o aplicaciones, las cuales son accedidas por los clientes utilizando un navegador que se comunica con el servidor utilizando el protocolo HTTP (*Hypertext Transfer Protocol*), este se mantiene a la espera de peticiones y le responde con el contenido según sea solicitado. (17)

Los servidores *web* permitirán alojar y desplegar en estos la propuesta de solución, dándole mayor seguridad y disponibilidad en todo momento al repositorio, siendo el utilizado apache en su versión 2.4.

### **Apache**

Servidor *web* de distribución libre y de código abierto, siendo uno de los servidores más populares y utilizados en el mundo, permitiéndose ejecutar en múltiples sistemas operativos. Es desarrollado y mantenido por una comunidad abierta de desarrolladores bajo el auspicio de la *Apache Software Foundation*.

Características según *OpenSuse* (18):

- Es un servidor *web* flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP.
- Multiplataforma.

- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API<sup>5</sup> de programación de módulos, para el desarrollo de módulos específicos.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

### 1.3.8 Herramientas a utilizar

#### NetBeans IDE

NetBeans IDE es un Entorno de Desarrollo Integrado (IDE), modular, de base estándar (normalizado) y escrito en el lenguaje de programación Java. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general (framework) para compilar cualquier tipo de aplicación (19). El IDE soporta varios lenguajes de Java, C / C ++, XML y HTML, PHP, Groovy, Javadoc, JavaScript y JSP. Puede instalarse en varios sistemas operativos: Windows, Linux o Mac OS.

Ofrece diferentes vistas de los datos, desde varias ventanas de proyectos a herramientas útiles para la creación de aplicaciones y su gestión de manera eficiente, lo que le permite desglosar sus datos de forma rápida y sencilla. También proporciona plantillas de código, consejos de codificación y herramientas de refactorización. Posee APIs (del inglés *Application Programming Interface*) bien documentadas y una comunidad extensa y activa de colaboradores (20). En la presente solución de investigación se utiliza esta en su versión 8.0.

Se selecciona el NetBeans IDE para compilar cualquier tipo de aplicación, ya que este soporta los lenguajes que se utilizarán para el desarrollo de la aplicación. Además, este es un IDE de código abierto y es multiplataforma.

#### Visual Paradigm para UML

Visual Paradigm es una herramienta CASE<sup>6</sup> profesional para el desarrollo de aplicaciones utilizando modelado UML, la cual es ideal para una amplia gama de personas interesadas

---

<sup>5</sup> **API:** Interfaz de programación de aplicaciones o del inglés *Application Programming Interface*.

<sup>6</sup> **CASE:** Ingeniería de Software Asistida por Computador o del inglés *Computer Aided Software Engineering*.

como ingenieros de *software*, analistas de sistemas y arquitectos de sistemas interesados en construir sistemas de *software* fiables y estables en el desarrollo orientado a objetos.

Emplea las últimas notaciones de UML, ingeniería inversa, generación de código, importación de *Rational Rose*, exportación e importación XML. Además, brinda una gran gama de posibilidades para desarrolladores y analistas tales como: diseñar la documentación del sistema con plantillas de diseño y estimar las consecuencias de los cambios con los diagramas de análisis de impacto, como la matriz y el diagrama de análisis, respectivamente. También soporta aplicaciones *web*, exporta en formato HTML, está disponible en varios idiomas y es fácil de instalar y de actualizar. (21)

Se selecciona la herramienta Visual Paradigm para realizar el modelado de la aplicación, facilitando así el desarrollo del capítulo 2 del presente documento.

### **pgAdmin3**

Como se expuso en el acápite 1.5.4 referente a los gestores de base de datos se eligió el uso del SGBD PostgreSQL, es por eso que sólo se analizará en el presente sub epígrafe las características de la herramienta que el portal oficial de la compañía define como idóneo para la administración de las bases de datos creadas sobre este SGBD: PgAdmin3.

Esta aplicación puede ser utilizada en los sistemas operativos Linux, FreeBSD, Solaris, MacOSX y Windows para administrar el SGBD PostgreSQL. Es de amplio uso debido a que está orientada a las necesidades de todos los usuarios, desde la escritura de simples consultas SQL a la elaboración de bases de datos complejas. La aplicación también incluye un editor de la sintaxis SQL, un editor de código del lado del servidor, un agente para la programación de tareas «SQL/batch/shell», soporte para el motor de replicación Slony-I, entre otras cosas. La conexión del servidor se puede realizar mediante TCP/IP<sup>7</sup> y puede ser cifrado mediante SSL<sup>8</sup> por seguridad. No se requieren controladores adicionales para comunicarse con la base de datos del servidor. Una característica interesante de pgAdmin3 es que posee la característica de reproducir la(s) sentencia(s) SQL correspondiente(s) a la acción realizada, lo que hace que, además de una herramienta muy útil, sea a la vez

---

<sup>7</sup> **TCP/IP**: Protocolo de control de transmisión/ Protocolo de Internet o del inglés *Transmission Control Protocol/ Internet Protocol*.

<sup>8</sup> **SSL**: Capa de conexiones seguras o del inglés *Secure Sockets Layer*.

didáctica. También incorpora funcionalidades para realizar consultas, examinar su ejecución y trabajar con los datos. (22)

### **1.3.9 Técnicas de Inteligencia Artificial empleadas para el proceso de toma de decisiones.**

A continuación se realiza un estudio de los diferentes Sistemas Basados en Conocimientos existentes, con el fin de identificar cual puede ser empleado para dar solución al problema de la investigación.

#### **Sistema Basado en Reglas**

Los Sistemas Basados en Reglas (SBR) son los más conocidos de los Sistemas Basados en el Conocimiento (SBC), en los que la forma de representación del conocimiento usado son las reglas de producción y como método de inferencia utiliza la regla de *modus ponens*.

Las reglas utilizan un formato *IF - THEN* para representar el conocimiento, la parte *IF* de una regla es una condición (también llamada premisa o antecedente), y la parte *THEN* de la regla (también llamada acción, conclusión o consecuente) permite inferir un conjunto de hechos nuevos si se verifican las condiciones establecidas en la parte *IF*. (23)

Los SBR son frecuentemente confundidos con sistemas lógicos; sin embargo, estos se definen en dos ideas principales: (23)

- Los SBR son generalmente no monotónicos, es decir los hechos pueden variar su veracidad durante el proceso de razonamiento.
- Los SBR aceptan incertidumbre en el proceso deductivo.

#### **Razonamiento Basado en Casos**

El Razonamiento Basado en Casos (RBC) representa un nuevo método para resolver problemas no estructurados, en el cual el razonamiento se realiza a partir de una memoria asociativa que usa un algoritmo para determinar una medida de semejanza entre dos objetos. En este paradigma la base del comportamiento inteligente de un sistema radica en recordar situaciones similares existentes en el pasado. Debe destacarse que es una técnica, en la cual la memoria se sitúa como fundamento de la IA y más concretamente de los SBC. (23)

El RBC denota un método donde la solución de un nuevo problema se realiza a partir de las soluciones conocidas para un conjunto de problemas previamente resueltos (o no

resueltos) del dominio de aplicación. Este método se distingue por el hecho de utilizar directamente la información almacenada en la memoria del sistema sobre los problemas o casos ya resueltos o no. Los sistemas que emplean el RBC usan una memoria permanente en lugar de alguna forma de base de conocimientos en la cual se almacene de forma explícita el conocimiento sobre el dominio de aplicación en forma de estructuras conceptuales, reglas de producción u otra forma de representación del conocimiento. (23)

### **1.3.10 Selección de la técnica de Inteligencia Artificial como apoyo en el proceso de toma de decisiones**

Se selecciona los Sistemas Basados en Reglas ya que esta es la que cumple de mejor manera con las necesidades reales para el desarrollo del sistema. Pues no se cuenta además con muchos procesos realizados con anterioridad, lo cual permitiría utilizar el RBC. Teniendo en cuenta lo antes expresado se empleará un SBR haciendo uso de la inferencia como principal arma de los mismos. El uso de esta técnica aportará al sistema una mayor robustez, mientras que a los especialistas que interactuarán con el mismo una mayor rapidez en la realización de la selección del modelo de negocio más adecuado para comercializar los productos del centro.

### **1.3.11 Metodología de desarrollo de *software*.**

La metodología de desarrollo del *software* constituye un proceso donde se definen técnicas y procedimientos para llevar a cabo el desarrollo de *software*. No existe hasta el momento una metodología que sea utilizada de forma universal. Cada una con sus características propias, pero en todas, el propósito es el mismo y es que el proceso sea configurable.

Cada metodología con sus diferencias en el grado de detalle en la manera en que gestionan con precisión los artefactos, roles y actividades del proyecto. Pero de manera general se puede encontrar en dos grandes grupos: metodologías tradicionales (o también denominadas Metodologías Pesadas, o Peso Pesado) y metodologías ágiles.

Las tradicionales establecen durante todo el proceso de desarrollo un mayor énfasis en la planificación y control del proyecto, y en la especificación precisa de los requisitos y el modelado. Por otro lado, las ágiles están más orientadas a la generación de código con ciclos muy cortos de desarrollo, enfocándose a equipos de desarrollo pequeños, haciendo especial hincapié en aspectos humanos asociados al trabajo en equipo e involucrando activamente al cliente en el proceso. (24)

Teniendo en cuenta lo anterior y la necesidad de adaptarse a las exigencias y condiciones del cliente en cuanto a ahorro de tiempo y costos en el ciclo de desarrollo y entrega de la solución con calidad, van a permitir que se profundice más en el estudio de las metodologías ágiles por encima de las robustas.

### AUP-UCI

El Proceso Unificado Ágil (AUP, por sus siglas en inglés) es una versión simplificada del Proceso Racional Unificado (RUP, por sus iniciales en inglés). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de *software* de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP.

Producto a la no existencia de una metodología de *software* universal se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Dentro de las técnicas ágiles que implementa esta nueva variación y que se mantienen válidas en su antecesor, se pueden mencionar. (25)

- ✓ Desarrollo dirigido por pruebas.
- ✓ Modelado ágil.
- ✓ Gestión de cambios ágil.
- ✓ Refactorización de base de datos.

El uso de la técnica de modelado ágil permite encapsular los requisitos funcionales en Historias de Usuario (HU) o descripción de requisitos por procesos.

A continuación, se presentan las fases propuestas por dicha metodología durante el proceso de creación de *software*.

**Tabla 1** Fases de la metodología AUP-UCI.

Fases	Objetivos
<b>Inicio</b>	<ul style="list-style-type: none"> <li>✓ Planeación del proyecto.</li> <li>✓ Estudio inicial de la organización cliente.</li> <li>✓ Alcance del proyecto.</li> <li>✓ Estimaciones de tiempo, esfuerzo y costo.</li> </ul>
<b>Ejecución</b>	<ul style="list-style-type: none"> <li>✓ Ejecución de las actividades requeridas para desarrollar el <i>software</i>.</li> <li>✓ Ajuste de los planes del proyecto.</li> <li>✓ Modelado del negocio.</li> <li>✓ Levantamiento de requisitos.</li> <li>✓ Elaboración la arquitectura y el diseño.</li> </ul>

	✓ Implementación y liberación del producto.
<b>Cierre</b>	✓ Análisis de los resultados y ejecución del proyecto. ✓ Cierre del proyecto.

Se selecciona esta metodología ágil ya que es la recomendada en la UCI para guiar los procesos de desarrollo de *software*.

#### **1.4 Conclusiones parciales**

Como parte del desarrollo del presente capítulo se determinan las siguientes conclusiones:

- ✓ El estudio de un grupo de definiciones asociados a la selección de los modelos de negocio permitieron un mejor entendimiento de la investigación.
- ✓ Fueron establecidas las herramientas y tecnologías necesarias para la correcta implementación de la informatización del proceso de selección de los modelos de negocio para la comercialización de los productos del CEDIN.
- ✓ La metodología AUP-UCI permitirá guiar todo el proceso de desarrollo, facilitando la implementación de la futura plataforma gracias a las facilidades que el uso de la misma implica.

## Capítulo 2: Análisis y diseño de la propuesta de solución

### Introducción

En el presente capítulo se realiza la descripción de la propuesta de solución y se detallan los principales requerimientos de la aplicación. Además, se explica el uso de los patrones de diseño empleados para la implementación del sistema y el patrón arquitectónico a aplicar en el mismo. Por otra parte, se describen las historias de usuarios para cada requisito funcional identificado. Por último, se diseñan los diagramas de clases de diseño y el modelo de base de datos.

### 2.1 Descripción de la propuesta de solución

La propuesta de solución consiste en una plataforma *web* que permitirá la gestión de los procesos relacionados con la selección de los modelos de negocio en la comercialización del CEDIN. El mismo contará con un sistema de autenticación basado en roles (Administrador, Especialista Comercial y Jefe de Centro). El Administrador es el encargado de gestionar los usuarios que interactúan con el sistema, asignándole los roles, pues se hace uso del Protocolo Ligero de Acceso a Directorios (LDAP, por sus siglas en inglés) para la autenticación en el mismo.

Por otra parte el Especialista Comercial será el encargado de realizar el proceso de selección de los modelos de negocio a emplear en la comercialización de los productos del centro. Para ello debe gestionar (Insertar, Modificar y Eliminar) los productos tanto del centro como los de la competencia, los segmentos de mercado a los cuales está dirigido el producto a comercializar y los modelos de negocio a emplear para dicha actividad. También debe gestionar los fabricantes (competencia) de los productos similares a los desarrollados en el centro. Además, el Especialista Comercial debe definir el costo del producto a comercializar. Para ello tiene que especificar las actividades, la cantidad de horas hombre dedicada a dicha actividad y el costo de la misma. Una vez que se define el costo del producto, se debe realizar un estudio de los precios de la competencia y una comparación entre los todos los productos.

También el sistema permitirá que cada proceso realizado (Reporte) se exporte en formato PDF<sup>9</sup>, así como descargar dicho documento. Además el reporte puede ser editado por el especialista comercial que realiza el mismo. El sistema enviará una notificación al Jefe de

---

<sup>9</sup> Documento de Formato Portable o del inglés *Portable Document Format*.

Centro cuando el Especialista Comercial le envíe el reporte realizado para su previa aprobación.

Para los productos que la Universidad de las Ciencias Informáticas comercializa se ha definido una política de estrategia marcaría que no será empleada en esta plataforma, debido a que la propuesta de solución no tiene fines comerciales.

### 2.1.1 Descripción del empleo del Sistema Basado en Reglas en la propuesta de solución

El SBR se empleará para realizar la comparación entre los productos de la competencia y el producto a comercializar, este se evidenciará mediante una matriz donde las filas serán los criterios de medidas a tener en cuenta (establecidos previamente por el criterio de los expertos) y las columnas las constituirán los productos. En las posiciones que se forman al unir una fila con una columna en la matriz se pondrá automáticamente un valor binario: **Si** cuando el producto cumple con dicha medida o **No** en caso contrario. Finalmente el sistema devuelve los modelos de los productos con mayor número de coincidencias (valores positivos). De existir varios productos con igual número de coincidencias el sistema realiza una nueva matriz de comparación entre los productos, pero esta vez teniendo en cuenta los segmentos de mercado a los cuales están dirigidos estos. A continuación se muestra un ejemplo de cómo se evidencia la matriz de comparación en la plataforma:

Sistemas	InfoPerf	Well Logger	Well View	Lowis
Gestión de alarmas	no	si	si	no
Generación de reportes	no	no	no	no
Adquisición de datos	no	no	no	no
Registro Histórico de variables	no	no	no	no
Drivers de comunicación	no	no	si	no
Gestores de Base de Datos	no	no	no	si
Comunicación con otros sistemas	no	si	no	no
Acceso mediante navegador Web	no	no	no	no
Control de usuario	no	no	no	no

**Figura 2** Tabla de comparación realizada por la plataforma SESMO para el producto SCADA SAINUX. Fuente: Sistema SESMO.

## 2.2 Actores del sistema

La aplicación va dirigida al GIE del CEDIN para ser empleada en el proceso de selección de los modelos de negocio para la comercialización de los productos de dicho centro. A continuación, se describen los actores que interactuarán con dicha aplicación *web*:

**Tabla 2** Actores del sistema. Fuente: Elaboración propia.

Actor	Rol	Descripción
<b>Administrador</b>	ROLE_SUPER_ADMIN	Es el encargado(a) de llevar el control total de la aplicación, además puede realizar el proceso de manera íntegra.
<b>Especialista Comercial</b>	ROLE_SUBORDINADO	Es el encargado(a) de todos los procesos relacionados con la selección de los modelos de negocio para la comercialización de los productos del centro.
<b>Jefe de Centro</b>	ROLE_JEFE	Es el encargado(a) de la revisión modelo de negocio seleccionado para la comercialización de los productos del centro. Además, puede gestionar todos los elementos del proceso de selección excepto los reportes.

## 2.3 Requisitos del sistema

“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”. (26)

### 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” (26). Para el desarrollo de la aplicación se levantaron los siguientes requisitos funcionales (RF):

Tabla 3 Requisitos Funcionales del sistema.

No. RF	Nombre del RF	Descripción
RF. 1	<b>Autenticar Usuario.</b>	El sistema debe permitir la autenticación de los usuarios haciendo uso de los servicios LDAP, especificando su usuario y contraseña.
RF. 2	<b>Insertar Usuario.</b>	El sistema debe permitir al administrador la inserción de nuevos usuarios en la base de datos.
RF. 3	<b>Modificar Usuario.</b>	El sistema debe permitir al administrador modificar el rol de un usuario existente en la base de datos.
RF. 4	<b>Eliminar Usuario.</b>	El sistema debe permitir al administrador eliminar usuarios existentes en la base de datos del mismo.
RF. 5	<b>Insertar Producto.</b>	El sistema debe permitir al especialista comercial la inserción de nuevos productos a la base de datos.
RF. 6	<b>Modificar Producto.</b>	El sistema debe permitir al especialista comercial la modificación de los atributos de los productos existentes en la base de datos.
RF. 7	<b>Eliminar Producto.</b>	El sistema debe permitir al especialista comercial eliminar productos existentes en la base de datos.
RF. 8	<b>Insertar Segmento de Mercado.</b>	El sistema debe permitir al especialista comercial la inserción de nuevos segmentos de mercado a la base de datos del mismo.
RF. 9	<b>Modificar Segmento de Mercado.</b>	El sistema debe permitir la modificación de los atributos de los segmentos de mercado almacenados en su base de datos.
RF. 10	<b>Eliminar Segmento de Mercado.</b>	El sistema debe permitir al especialista comercial eliminar los segmentos de mercado existentes en su base de datos.
RF. 11	<b>Insertar Fabricante.</b>	El sistema debe permitir al especialista comercial la inserción de nuevos fabricantes de productos a la base de datos del mismo.

<b>RF. 12</b>	<b>Modificar Fabricante.</b>	El sistema debe permitir la modificación de los atributos de los fabricantes almacenados en su base de datos.
<b>RF. 13</b>	<b>Eliminar Fabricante.</b>	El sistema debe permitir al especialista comercial eliminar los fabricantes existentes en su base de datos.
<b>RF. 14</b>	<b>Insertar Modelo de Negocio.</b>	El sistema debe permitir al especialista comercial la inserción de nuevos modelos de negocio a la base de datos del mismo.
<b>RF. 15</b>	<b>Modificar Modelo de Negocio.</b>	El sistema debe permitir la modificación de los atributos de los modelos de negocio almacenados en su base de datos.
<b>RF. 16</b>	<b>Eliminar Modelo de Negocio.</b>	El sistema debe permitir al especialista comercial eliminar los modelos de negocio existentes en su base de datos.
<b>RF. 17</b>	<b>Insertar Actividad.</b>	El sistema debe permitir al especialista comercial la inserción de nuevas actividades a la base de datos del mismo.
<b>RF. 18</b>	<b>Modificar Actividad.</b>	El sistema debe permitir la modificación de los atributos de las actividades almacenadas en su base de datos.
<b>RF. 19</b>	<b>Eliminar Actividad.</b>	El sistema debe permitir al especialista comercial eliminar las actividades existentes en su base de datos.
<b>RF. 20</b>	<b>Insertar Medida.</b>	El sistema debe permitir al especialista comercial la inserción de nuevas medidas a la base de datos del mismo.
<b>RF. 21</b>	<b>Modificar Medida.</b>	El sistema debe permitir la modificación de los atributos de las medidas almacenadas en su base de datos.
<b>RF. 22</b>	<b>Eliminar Medida.</b>	El sistema debe permitir al especialista comercial eliminar las medidas existentes en su base de datos.

RF. 23	<b>Editar Reporte</b>	El sistema debe permitir al especialista comercial editar los reportes existentes en la BD.
RF. 24	<b>Generar Modelo de Negocio.</b>	El sistema debe permitir al especialista comercial seleccionar el modelo de negocio más adecuado para comercializar sus productos.
RF. 25	<b>Enviar Notificación.</b>	El sistema debe informar vía correo electrónico a al usuario con rol jefe de centro, cuando le sea enviado un reporte para su previa revisión.
RF. 26	<b>Exportar Reporte.</b>	El sistema debe permitir al especialista comercial exportar el proceso de selección del modelo de negocio a utilizar en formato PDF.
RF. 27	<b>Descargar Documentación.</b>	El sistema debe permitir al usuario descargar el documento PDF que contiene la selección.
RF. 28	<b>Mostrar Ayuda.</b>	El sistema debe brindar una ayuda a los usuarios para facilitar la interacción de estos con el mismo.
RF. 29	<b>Cerrar Sesión.</b>	El sistema debe permitir al usuario autenticado poder finalizar su sección de trabajo en cualquier momento y desde cualquier estación de trabajo.

### 2.3.2 Requisitos no funcionales

“Los requerimientos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluye restricciones de tiempo, sobre el proceso de desarrollo y estándares” (26). Para el desarrollo de la aplicación se capturaron los siguientes requisitos no funcionales (RnF):

**Tabla 4** Requisitos no Funcionales del sistema.

No. RnF	Tipo de RnF	Descripción
RnF. 1	<b>Usabilidad.</b>	<p><b>1.1</b> El sistema a desarrollar debe poseer un dominio de aplicación <i>web 2.0</i>.</p> <p><b>1.2</b> El sistema puede ser utilizado por personas que tengan un conocimiento básico en el manejo de las computadoras.</p>

<p><b>RnF. 2</b></p>	<p><b>Software.</b></p>	<p><u>Para el cliente:</u>  <b>2.1</b> La aplicación debe de correr en sistema operativo Windows XP o superior, o GNU Linux en cualquiera de sus distribuciones.  <b>2.2</b> Para interactuar con el sistema se debe utilizar el navegador (Mozilla Firefox 25 o superior, Google Chrome 30 o superior).  <u>Para el servidor:</u>  <b>2.3</b> El servidor debe tener sistema operativo Windows XP o GNU Linux en cualquiera de sus distribuciones.  <b>2.4</b> El sistema debe contar con el servidor de base de datos PostgreSQL 9.2 o superior.  <b>2.5</b> El sistema debe contar con un servidor con módulo PHP integrado (WAMP, XAMPP u otros).</p>
<p><b>RnF. 3</b></p>	<p><b>Hardware.</b></p>	<p><b>3.1</b> Capacidad del disco duro superior a los 5 Giga bytes.  <b>3.2</b> Se requiere un mínimo de 512 Mega bytes de RAM y 1.6 GHz de velocidad de procesamiento.</p>
<p><b>RnF. 4</b></p>	<p><b>Apariencia o interfaz externa.</b></p>	<p><b>4.1</b> En el sistema las interfaces están sustentadas por los colores: azul, blanco, negro, cian y gris. Por otra parte, cuenta con un menú en el lateral izquierdo el cual recoge las funcionalidades específicas.  <b>4.2</b> El sistema debe contar con un diseño <i>responsive</i> (que se adapte a cualquier tipo de dispositivo como son: <i>smartphone, tablet</i>, entre otros.).</p>
<p><b>RnF. 5</b></p>	<p><b>Seguridad.</b></p>	<p><b>5.1</b> La contraseña de los usuarios debe ser encriptada haciendo uso del algoritmo de encriptación md5<sup>10</sup>.</p>

<sup>10</sup> **MD5:** Algoritmo del Resumen del Mensaje 5 o del inglés *Message-Digest Algorithm 5*.

		<b>5.2</b> El sistema debe 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.
<b>RnF. 6</b>	<b>Requisitos legales, de derecho de autor y otros.</b>	<b>6.1</b> Los requisitos legales, de derecho de autor y otros se establecen según el centro de calidad de la UCI y el centro de desarrollo en cuestión, CEDIN.

## 2.4 Historias de usuarios

Una Historia de Usuario (HU) es una representación de un requisito de *software* escrito de una o dos frases utilizando el lenguaje común del usuario. Las HU son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos. Además, se consideran una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir mucho tiempo para administrarlos, permitiendo responder rápidamente a los requisitos cambiantes. (27)

Durante el diseño de la propuesta de solución se identificaron 29 historia de usuario que responden a las diferentes funcionalidades solicitadas por el cliente y presentan una descripción para que el desarrollador conozca su posterior implementación. A continuación, se describen algunas de las HU del sistema:

**Tabla 5** Historia de Usuario "Insertar Segmento de Mercado".

<b>Historia de Usuario</b>	
<b>Número 8</b>	<b>Nombre de la Historia de Usuario:</b> Insertar Segmento de Mercado.
<b>Cantidad de modificaciones a la Historia de Usuario:</b> Ninguna.	
<b>Usuario:</b> Especialista Comercial.	<b>Iteración asignada:</b> 1
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto

**Descripción:** El sistema debe permitir al especialista comercial la inserción de nuevos segmentos de mercado a la base de datos del mismo. Para ello el actor involucrado debe de llenar el formulario que el sistema le provee. Dicho formulario cuenta con los siguientes campos: “**Nombre**” y “**Descripción**”. Donde el campo nombre se refiere al nombre del segmento de mercado al cual está dirigido el producto, este campo solo admite letras (*string*, 255) y el campo descripción se refiere a la descripción de dicho segmento y admite valores alfanuméricos (*text*). Una vez llenados dichos campos el actor debe de hacer clic en el botón “**Aceptar**” y automáticamente el sistema muestra un mensaje de operación satisfactoria. En caso de presentar algún error en el contenido introducido por el actor en el formulario, se muestra un mensaje de error en el campo donde se cometió el mismo y se mantiene en dicho formulario.

**Prototipo:**

**Tabla 6** Historia de Usuario "Modificar Segmento de Mercado".

Historia de Usuario	
<b>Número 9</b>	<b>Nombre de la Historia de Usuario:</b> Modificar Segmento de Mercado.
<b>Cantidad de modificaciones a la Historia de Usuario:</b> Ninguna.	
<b>Usuario:</b> Especialista Comercial.	<b>Iteración asignada:</b> 1
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Descripción:</b> El sistema debe permitir la modificación de los atributos de los segmentos de mercado almacenados en su base de datos. Esto ocurrirá en caso de que se necesite cambiar la información sobre el segmento de mercado ya existente. Para ello debe generarse el formulario del segmento de mercado 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 de operación satisfactoria.

**Prototipo:**

≡ Datos del segmento

Nombre del segmento:

Descripción del segmento:

Cancelar
Aceptar

**Tabla 7** Historia de Usuario "Eliminar Segmento de Mercado".

Historia de Usuario							
<b>Número 10</b>	<b>Nombre de la Historia de Usuario:</b> Eliminar Segmento de Mercado.						
<b>Cantidad de modificaciones a la Historia de Usuario:</b> Ninguna.							
<b>Usuario:</b> Especialista Comercial.	<b>Iteración asignada:</b> 2						
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio						
<b>Descripción:</b> El sistema debe permitir al especialista comercial eliminar los segmentos de mercado existentes en su base de datos. Para ello debe de seleccionar el segmento de mercado a eliminar y ejecutar dicha acción. El sistema debe de mostrar un mensaje de operación satisfactoria.							
<b>Prototipo:</b>							
<div style="background-color: #00c853; color: white; padding: 5px; margin-bottom: 10px;">Listado de segmentos</div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Nombre</th> <th style="width: 60%;">Descripción</th> <th style="width: 25%;">Gestión</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">Industria Eléctrica</td> <td style="padding: 5px;">Los rápidos cambios en la industria eléctrica impulsados por los competitivos niveles de productividad, eficiencia y de calidad de servicio, requieren de una constante innovación a la hora de resolver la automatización de Unidades Generadoras y Subestación Eléctrica (S/E), lo cual puede significar difíciles desafíos.</td> <td style="padding: 5px; text-align: right;"> <a href="#" style="color: #0070c0; text-decoration: none;">✎ Modificar</a>  <a href="#" style="color: #0070c0; text-decoration: none;">🗑 Eliminar</a> </td> </tr> </tbody> </table>		Nombre	Descripción	Gestión	Industria Eléctrica	Los rápidos cambios en la industria eléctrica impulsados por los competitivos niveles de productividad, eficiencia y de calidad de servicio, requieren de una constante innovación a la hora de resolver la automatización de Unidades Generadoras y Subestación Eléctrica (S/E), lo cual puede significar difíciles desafíos.	<a href="#" style="color: #0070c0; text-decoration: none;">✎ Modificar</a> <a href="#" style="color: #0070c0; text-decoration: none;">🗑 Eliminar</a>
Nombre	Descripción	Gestión					
Industria Eléctrica	Los rápidos cambios en la industria eléctrica impulsados por los competitivos niveles de productividad, eficiencia y de calidad de servicio, requieren de una constante innovación a la hora de resolver la automatización de Unidades Generadoras y Subestación Eléctrica (S/E), lo cual puede significar difíciles desafíos.	<a href="#" style="color: #0070c0; text-decoration: none;">✎ Modificar</a> <a href="#" style="color: #0070c0; text-decoration: none;">🗑 Eliminar</a>					

**Tabla 8** Historia de Usuario "Insertar Modelo de Negocio".

Historia de Usuario	
Número 14	Nombre de la Historia de Usuario: Insertar Modelo de Negocio.
Cantidad de modificaciones a la Historia de Usuario: Ninguna.	
Usuario: Especialista Comercial.	Iteración asignada: 1
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
<p><b>Descripción:</b> El sistema debe permitir al especialista comercial la inserción de nuevos modelos de negocio a la base de datos del mismo. Para ello el actor involucrado debe de llenar el formulario que el sistema le provee. Dicho formulario cuenta con los siguientes campos: “<b>Nombre</b>” y “<b>Descripción</b>”. Donde el campo nombre se refiere al nombre del modelo de negocio empleado para comercializar el producto, este campo solo admite letras (<i>string</i>, 255), el campo descripción se refiere a la descripción del mismo y admite valores alfanuméricos (<i>text</i>). Una vez llenados dichos campos el actor debe de hacer clic en el botón “<b>Aceptar</b>” y automáticamente el sistema muestra un mensaje de operación satisfactoria. En caso de presentar algún error en el contenido introducido por el actor en el formulario, se muestra un mensaje de error en el campo donde se cometió el mismo y se mantiene en dicho formulario.</p>	
Prototipo:	 <p>The screenshot shows a web interface with a green header bar containing a hamburger menu icon and the text 'Datos del modelo'. Below the header, there are two input fields. The first is labeled 'Nombre del modelo:' and contains the placeholder text 'Entre el nombre del modelo'. The second is labeled 'Descripción del modelo:' and contains the placeholder text 'Entre una breve descripción'. At the bottom right of the form area, there is a green button labeled 'Aceptar'.</p>

**Tabla 9** Historia de Usuario "Modificar Modelo de Negocio".

Historia de Usuario	
<b>Número 15</b>	<b>Nombre de la Historia de Usuario:</b> Modificar Modelo de Negocio.
<b>Cantidad de modificaciones a la Historia de Usuario:</b> Ninguna.	
<b>Usuario:</b> Especialista Comercial.	<b>Iteración asignada:</b> 1
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<p><b>Descripción:</b> El sistema debe permitir la modificación de los atributos de los modelos de negocio almacenados en su base de datos. Esto ocurrirá en caso de que se necesite cambiar la información sobre el modelo de negocio ya existente. Para ello debe generarse el formulario del modelo de negocio 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 de operación satisfactoria.</p>	
<b>Prototipo:</b>	 <p>The screenshot shows a web interface for modifying a business model. It features a green header bar with the text 'Datos del modelo'. Below the header, there are two input fields: 'Nombre del modelo:' with a placeholder 'Entre el nombre del modelo' and 'Descripción del modelo:' with a placeholder 'Entre una breve descripción'. At the bottom right of the form area, there is a green button labeled 'Aceptar'.</p>

**Tabla 10** Historia de Usuario "Eliminar Modelo de Negocio".

Historia de Usuario	
<b>Número 16</b>	<b>Nombre de la Historia de Usuario:</b> Eliminar Modelo de Negocio.
<b>Cantidad de modificaciones a la Historia de Usuario:</b> Ninguna.	

<b>Usuario:</b> Especialista Comercial.	<b>Iteración asignada:</b> 2						
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio						
<p><b>Descripción:</b> El sistema debe permitir al especialista comercial eliminar los modelos de negocio existentes en su base de datos. Para ello debe de seleccionar el modelo de negocio a eliminar y ejecutar dicha acción. El sistema debe de mostrar un mensaje de operación satisfactoria.</p>							
<p><b>Prototipo:</b></p> <div style="background-color: #00c853; color: white; padding: 5px; margin-bottom: 5px;">Listado de modelos de negocio</div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Nombre</th> <th style="width: 60%;">Descripción</th> <th style="width: 20%;">Gestión</th> </tr> </thead> <tbody> <tr> <td>Modelo de Comercialización de servicios asociados a un producto</td> <td>En este modelo, la empresa promueve un producto libre generalmente desarrollado por ella o en cuyo desarrollo participa activamente y una vez creado un mercado para este, la empresa comienza a vender los servicios generados en torno al producto como: consultoría, personalizaciones o adaptaciones, capacitación, mantenimiento, actualización y soporte.</td> <td style="text-align: right;"> <a href="#" style="color: #0070c0; text-decoration: none;">✎ Modificar</a>  <a href="#" style="color: #0070c0; text-decoration: none;">🗑 Eliminar</a> </td> </tr> </tbody> </table>		Nombre	Descripción	Gestión	Modelo de Comercialización de servicios asociados a un producto	En este modelo, la empresa promueve un producto libre generalmente desarrollado por ella o en cuyo desarrollo participa activamente y una vez creado un mercado para este, la empresa comienza a vender los servicios generados en torno al producto como: consultoría, personalizaciones o adaptaciones, capacitación, mantenimiento, actualización y soporte.	<a href="#" style="color: #0070c0; text-decoration: none;">✎ Modificar</a> <a href="#" style="color: #0070c0; text-decoration: none;">🗑 Eliminar</a>
Nombre	Descripción	Gestión					
Modelo de Comercialización de servicios asociados a un producto	En este modelo, la empresa promueve un producto libre generalmente desarrollado por ella o en cuyo desarrollo participa activamente y una vez creado un mercado para este, la empresa comienza a vender los servicios generados en torno al producto como: consultoría, personalizaciones o adaptaciones, capacitación, mantenimiento, actualización y soporte.	<a href="#" style="color: #0070c0; text-decoration: none;">✎ Modificar</a> <a href="#" style="color: #0070c0; text-decoration: none;">🗑 Eliminar</a>					

## 2.5 Arquitectura del sistema

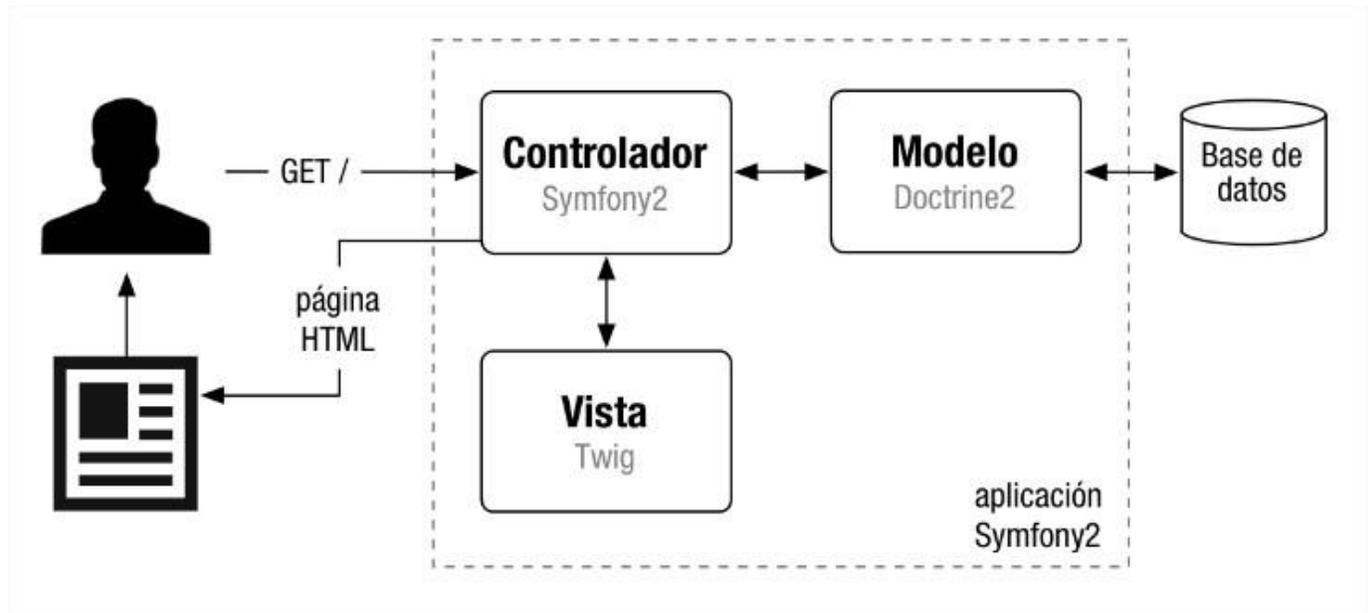
La arquitectura de un sistema de *software* no es más que la organización o estructura de los componentes importantes que interactúan en el mismo.

Durante el desarrollo de un *software* frecuentemente se encuentran distintos puntos de vista entre el equipo de trabajo responsable de la construcción de un sistema, la arquitectura es el instrumento encargado de unificar estas diferencias. La misma tiene como propósito principal brindar elementos que ayuden a la toma de decisiones y proporcionar un lenguaje común. La arquitectura de *software* para lograr esto se enfoca en seleccionar y combinar estilos y patrones arquitectónicos.

Para el desarrollo del sistema propuesto se seleccionó el patrón arquitectónico Modelo-Vista-Controlador (MVC), el cual permite un mejor soporte para futuras actualizaciones del producto, dividiendo la aplicación en tres componentes distintos: el modelo, la vista y el controlador de forma tal que los cambios ocurridos en una de estas abstracciones, no afecte a las demás. Actualmente, este patrón es muy utilizado para el desarrollo de aplicaciones *web* de gran alcance, permitiendo desarrollar proyectos más extensibles de cualquier aplicación basada en otro patrón. Se utiliza el patrón MVC ya que Symfony 2, basa su

funcionamiento interno en esta famosa arquitectura. A continuación, se detalla en qué consisten las tres abstracciones del patrón (28):

- ✓ **Modelo:** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- ✓ **Vista:** Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- ✓ **Controlador:** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de los botones del ratón, pulsaciones de teclas, entre otros. Los eventos son traducidos a solicitudes de servicios (*service requests*) para el modelo o la vista.



**Figura 3** Patrón Arquitectónico Modelo-Vista-Controlador para Symfony 2. (29)

## 2.6 Patrones de diseño del sistema

Para comenzar la implementación de un *software* no basta con definir la arquitectura, es necesario además establecer las directrices que permitan lograr un sistema bien estructurado, para así construir una solución eficaz. Estas directrices son llamadas Patrones de Diseño de *Software*. Para satisfacer las necesidades de la sociedad cada día se hace más necesario desarrollar un *software* de gran alcance y complejidad, en lo que son de gran utilidad los patrones de diseño empleados como mecanismos de reutilización.

Para la realización del sistema se emplearán cinco patrones de la familia de los GRASP<sup>11</sup>, estos son: Experto, Creador, Bajo Acoplamiento, Alta Cohesión y el Controlador. A continuación, se muestra la selección de los mismos:

Empleando el patrón **Experto** se garantiza que cada clase del sistema asuma las responsabilidades que le conciernen, según las funcionalidades que se quieren implementar y a partir de la información que posee; por lo que cada clase contendrá la información necesaria para cumplir su responsabilidad. Al utilizar el patrón **Creador** cada clase instancia y crea las clases que colaborarán con la misma para cumplir con su responsabilidad, tratando de lograr siempre un **Bajo acoplamiento** y una **Alta cohesión**. Con el cumplimiento de estos dos últimos patrones se logra que cada clase recurra solamente a las clases que son imprescindibles para su trabajo y tenga asociada las responsabilidades que le corresponden de acuerdo con su comportamiento. Además, al emplear el patrón **Controlador**, se estructura el sistema con una clase controladora para cada concepto importante del modelo de dominio para que sea esta quien, respetando las condiciones del patrón arquitectónico MVC, se encargue de los eventos y funcionalidades (crear, modificar, mostrar, eliminar) que representan dicho concepto.

---

<sup>11</sup> **GRASP**: Patrones generales para asignar responsabilidades o del inglés *General Responsibility Assignment Software Patterns*.



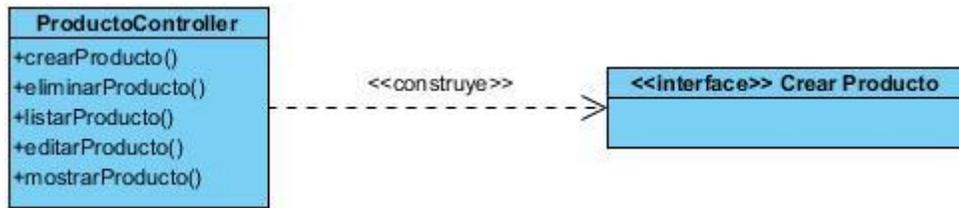


Figura 6 Diagrama que representa el uso del Patrón Controlador en el sistema.

## 2.7 Diagrama de clases

Un tipo de diagrama de vital importancia para la definición del sistema es el diagrama de clases. En este diagrama se representan las entidades de forma estática en forma de clases. Una clase puede contener atributos, propiedades y métodos (30). A continuación, se muestra un diagrama de clases del sistema:

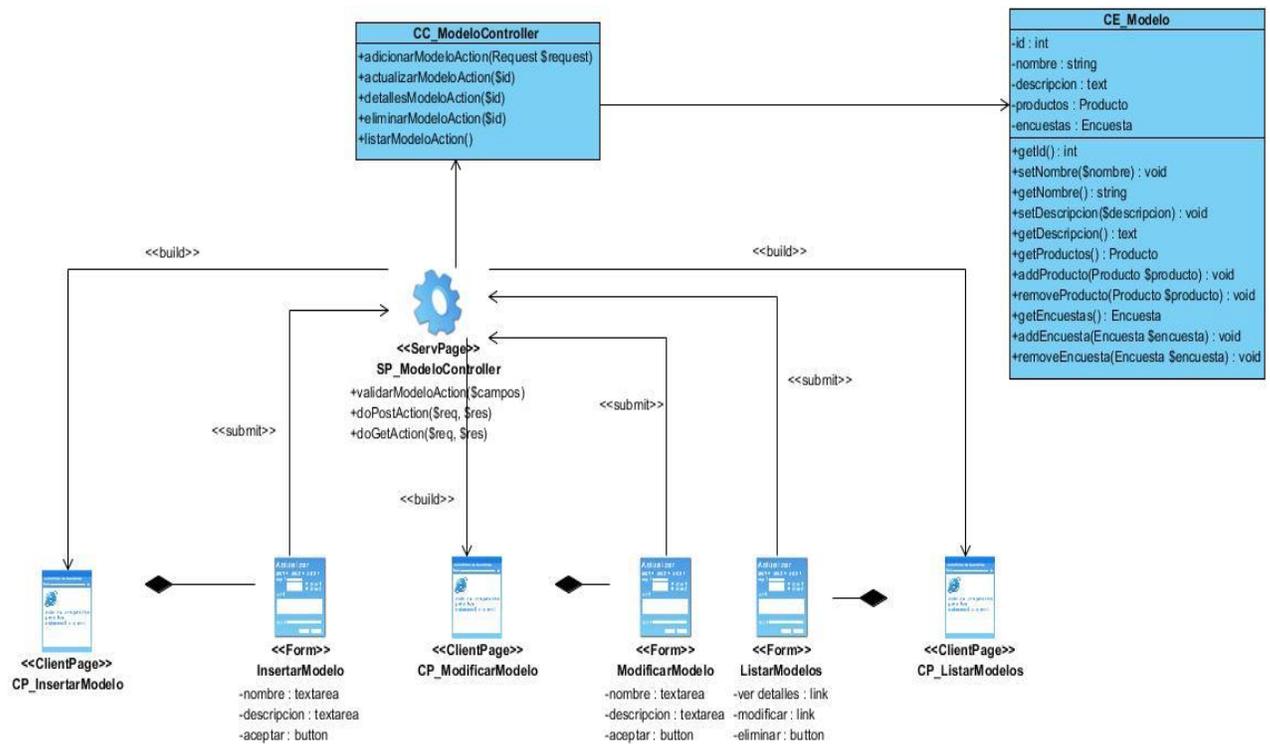


Figura 7 Diagrama de clases de los requisitos Insertar, Modificar y Eliminar Modelos de Negocio.

## 2.8 Modelo de datos

El modelo de datos no es más que un sistema formal y abstracto que permite describir los datos de acuerdo con reglas y convenios predefinidos que podríamos decir que es un conjunto de concepto que permite describir, a distintos niveles de abstracción, la estructura de una base de datos (31). A continuación, se muestra el modelo de datos correspondiente a la solución propuesta.



## Capítulo 3: Implementación y pruebas a la solución propuesta

### Introducción

La concepción de la propuesta del sistema presentada en el capítulo anterior, ayuda al programador a entender mejor las funcionalidades que busca el cliente y, por tanto, ser capaz de llevarlas a código entendible por la computadora, o lo que es lo mismo implementar el sistema. Durante toda la fase de implementación y después de esta se desarrollan un conjunto de pruebas con el objetivo de asegurar la calidad de la aplicación y que la misma cumpla con todas las peticiones del cliente. Como principales elementos en este capítulo se definen el diagrama de componentes y el diagrama de despliegue. Además, de especificar los casos de pruebas aplicados a la solución desarrollada para validar su correcto funcionamiento.

### 3.1 Implementación

La implementación es un proceso que comienza cuando se crea una aplicación en el equipo de un desarrollador y termina cuando está instalada y lista para ejecutarse en el equipo de un usuario. Describe cómo los elementos del modelo de diseño se implementan en términos de componentes, en otras palabras, toma el resultado del modelo de diseño para generar el código final del sistema. Dicho código está determinado por el lenguaje de programación y tiene como objetivo llevar a cabo la implementación de cada una de las clases significativas del diseño.

#### 3.1.1 Estándares de código

Un estándar de codificación comprende todos los aspectos de la generación de código. Un código fuente completo debe de reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Los estándares de codificación permiten una mejor integración entre las líneas de producción y establecen pautas que conllevan a lograr un código más legible y reutilizable, de tal forma que se pueda aumentar su capacidad de almacenamiento a lo largo del tiempo. (32)

#### Nomenclatura de las clases

Los nombres de las clases comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará la notación **PascalCasing** (cada palabra comienza con letra mayúscula). Por ejemplo (**Modelo** o

**ModeloDeNegocio**). En el caso particular de las clases controladoras después del nombre llevan la palabra "Controller", por ejemplo **ProductoController**.

### Nomenclatura de las funciones

El nombre a emplear para las funciones se escribe con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará la notación **CamelCasing** (es parecido al *PascalCasing* con la excepción que la letra inicial del identificador debe estar en minúscula), y con solo leerlo se sabe el propósito de la misma. Por ejemplo (**adicionarModelo** o **adicionarModeloDeNegocio**).

### Nomenclatura de las variables

El nombre de las variables se escribe con la primera letra en minúscula, si es un nombre compuesto se empleará la notación de **CamelCasing** descrita anteriormente. Siempre las variables llevarán delante del nombre el símbolo de peso ("\$"). Por ejemplo (**\$nombre** o **\$nombreProducto**)

### Nomenclatura de los comentarios

Los comentarios deben ser precisos de forma tal que se entienda el propósito de lo que se está desarrollando. Siempre se pondrán encima de la función a la cual hace alusión dicho comentario. Por ejemplo:

```
/**
```

```
* Crea a nueva entidad Encuesta descripción de la función
```

```
* @Route("/new", name="encuesta_new") ruta de donde accede a los datos necesarios para la implementación de la función
```

```
* @Method({"GET", "POST"}) método por el cual captura los datos.
```

```
*/
```

### **3.1.2 Diagrama de componentes**

Este tipo de diagrama representa cómo un sistema de *software* es dividido en componentes, los cuáles pueden ser archivos, bibliotecas, módulos, paquetes o ejecutables y muestra el

comportamiento de los servicios que estos componentes proporcionan y utilizan a través de las interfaces. A continuación, se muestra el diagrama de componentes del sistema:

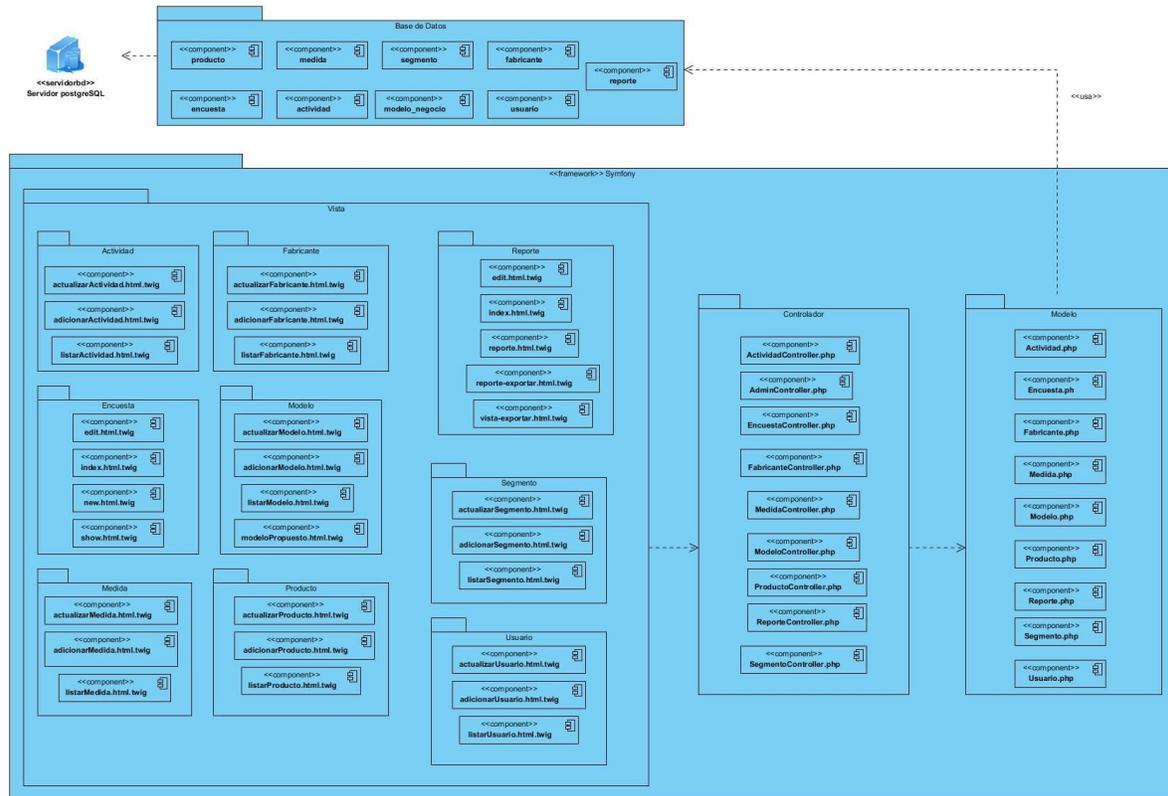
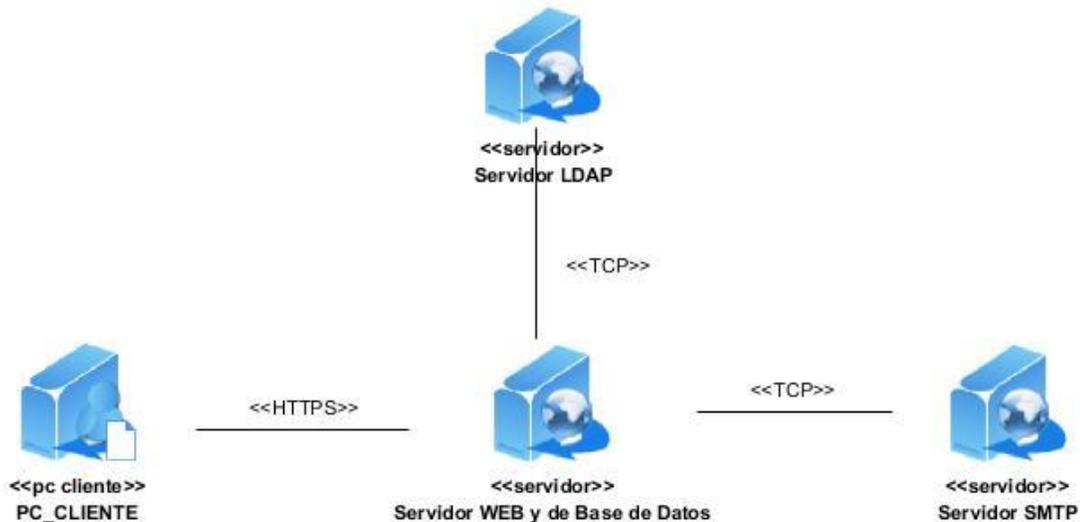


Figura 9 Diagrama de componentes de la aplicación.

### 3.1.3 Diagrama de despliegue

El diagrama de despliegue como se muestra en la figura 9, es un recurso que se utiliza para emular las condiciones físicas en las que va a estar desplegado el sistema. Este diagrama es utilizado como base para la realización de las actividades de diseño e implementación. Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo hardware similar. Dichos nodos poseen relaciones que representan medios de comunicación entre ellos, tales como, Internet, Intranet, bus y similares. A continuación, se muestra el diagrama de despliegue del sistema propuesto.



**Figura 10** Diagrama de despliegue de la aplicación.

El diagrama establece la relación existente entre los nodos:

- ✓ **PC\_CLIENTE:** es el espacio de trabajo por el cual el usuario interactúa con el sistema. Esta computadora debe de tener instalado el sistema operativo Windows o GNU/ Linux, y como navegador *web* Google Chrome o Mozilla Firefox 2.0 como mínimo.
- ✓ **Servidor WEB y de BD:** se utilizará para publicar la aplicación, empleando el servidor *Web* Apache más el lenguaje PHP. Además, se utilizará PostgreSQL como Base de Datos (BD) que permitirá almacenar toda la información de la aplicación.
- ✓ **Servidor LDAP:** son servicios web que brinda la UCI para obtener información del personal de la universidad ya sean estudiantes, trabajadores o profesores que son autenticados en el dominio uci.cu mediante LDAP.
- ✓ **Servidor SMTP:** es el espacio utilizado para el intercambio de mensajes de correos entre computadoras u otros dispositivos. Este se empleará para realizar el envío de correo mediante la aplicación al Jefe de Centro.

**Conexión HTTPS:** es el protocolo que se usa en cada transacción de datos en la web, utiliza un cifrado basado en SSL y así crear un canal de transferencia cifrado con el que aumenta la seguridad en el tráfico de información en comparación al protocolo HTTP común.

**Conexión TCP:** es el protocolo utilizado entre servidores. Este elemento de la arquitectura representa un tipo de comunicación que permite a los anfitriones establecer una conexión e intercambiar datos.

### 3.2 Pruebas de *software*

“Las pruebas constituyen el último bastión desde el que se puede evaluar la calidad y, de forma más pragmática descubrir errores.” (33)

Luego de generado el código de fuente, es necesario realizarle pruebas para detectar y corregir la mayor cantidad de errores posibles con el objetivo de brindar una solución con la calidad requerida. Existen diferentes tipos de pruebas, las cuales tienen como objetivo detectar alguna anomalía en los resultados. En su mayoría están orientadas a comprobar las funcionalidades y lógica del negocio, verificando que el sistema se comporte a la altura de las especificaciones que pide el cliente.

#### Métodos de las pruebas

- ✓ Pruebas de caja negra: pruebas que se llevan a cabo sobre la interfaz del *software*. El objetivo es demostrar que las funciones del *software* son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene (no se analiza el código).
- ✓ Pruebas de caja blanca: pruebas que se producen sobre el código del *software*. Es considerada como uno de los tipos de pruebas más importantes que se le aplican a los programas informáticos, logrando como resultado que disminuyan en un gran porcentaje el número de errores existentes en los sistemas y por ende, se logre una mayor calidad y confiabilidad.

Cada uno de estos métodos cuenta con distintos niveles de pruebas que a continuación se mencionan:

- ✓ Prueba de desarrollador.
- ✓ Prueba independiente.
- ✓ Prueba de unidad.
- ✓ Prueba de integración.
- ✓ Prueba de sistema.
- ✓ Prueba de aceptación.

### 3.2.1 Casos de prueba de Aceptación

De los niveles de pruebas de *software* existentes mencionados con anterioridad se selecciona para comprobar la calidad de la aplicación desarrollada las pruebas de aceptación, pues este es el tipo de pruebas recomendado por la metodología de desarrollo empleada. Es válido aclarar que estas pruebas las realiza el propio cliente acompañado del equipo de desarrollo y se orientan a las funcionalidades del sistema. Dentro de las pruebas de aceptación se utilizó la técnica de partición de equivalencia. Este tipo de pruebas genera como artefacto los Casos de Pruebas (CP). “Los casos de pruebas son actividades en las cuales un sistema o componente es ejecutado bajo condiciones o requerimientos especificados, permitiendo encontrar y documentar los defectos que puedan afectar la calidad del *software*”. (34)

A continuación, se muestran una parte de los CP correspondientes a las pruebas de aceptación realizados a la aplicación:

**Tabla 11** Caso de prueba de aceptación para la HU "Insertar Segmento de Mercado".

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> R8_P1	<b>Historia de Usuario:</b> 8
<b>Nombre:</b> Insertar Segmento de Mercado.	
<b>Descripción:</b> El sistema debe permitir al especialista comercial la inserción de nuevos segmentos de mercado a la base de datos del mismo.	
<b>Condiciones de Ejecución:</b> Usuario con permisos de Especialista Comercial debe estar autenticado.	
<b>Entrada/Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. Clic en el botón “<b>Añadir Segmento</b>” del módulo Segmentos de Mercado ubicado en el menú lateral izquierdo del sistema.</li> <li>2. Introducir los datos en los campos (Nombre, Descripción). Juego de datos (“<b>Industria 1</b>”, “<b>Este segmento es muy bueno.</b>”).</li> <li>3. Clic en el botón “<b>Aceptar</b>”.</li> </ol>	
<b>Resultado esperado:</b> El segmento debe ser añadido a la base de datos de la aplicación. Se muestra un mensaje de operación satisfactoria.	

**Resultado obtenido:**



**Evaluación de la prueba:** Satisfactoria.

**Tabla 12** Caso de prueba de aceptación para la HU "Modificar Segmento de Mercado".

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> R9_P1	<b>Historia de Usuario:</b> 9
<b>Nombre:</b> Modificar Segmento de Mercado.	
<b>Descripción:</b> El Especialista comercial puede modificar los valores de los segmentos guardados en la base de datos del sistema.	
<b>Condiciones de Ejecución:</b> Usuario con permisos de Especialista Comercial debe estar autenticado y deben existir segmentos en la base de datos del sistema.	
<b>Entrada/Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. Seleccionar la opción <b>“Gestionar Segmentos”</b> en el módulo Segmentos de Mercado que se encuentra en el menú lateral izquierdo.</li> <li>2. Seleccionar el segmento que desea modificar y de este la opción de modificar segmento.</li> <li>3. Introducir los datos del/los campos que se desean modificar (Nombre y Descripción). Juego de datos (<b>“Industria 2”</b>, <b>“Este segmento es muy bueno.”</b>).</li> <li>4. Clic en el botón <b>“Aceptar”</b>.</li> </ol>	
<b>Resultado esperado:</b> El sistema debe modificar en la base de datos los valores introducidos en el segmento. Se muestra un mensaje de operación satisfactoria.	
<b>Resultado obtenido:</b>	
<b>Evaluación de la prueba:</b> Satisfactoria.	

**Tabla 13** Caso de prueba de aceptación para la HU "Eliminar Segmento de Mercado".

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> R10_P1	<b>Historia de Usuario:</b> 10
<b>Nombre:</b> Eliminar Segmento de Mercado.	
<b>Descripción:</b> El Especialista comercial puede eliminar los segmentos existentes en la base de datos del sistema.	
<b>Condiciones de Ejecución:</b> Usuario con permisos de Especialista Comercial debe estar autenticado y deben existir segmentos en la base de datos del sistema.	
<b>Entrada/Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. Seleccionar la opción <b>"Gestionar Segmentos"</b> en el módulo Segmentos de Mercado que se encuentra en el menú lateral izquierdo.</li> <li>2. Seleccionar el segmento que desea eliminar.</li> <li>3. Clic en la opción <b>"Eliminar"</b>.</li> </ol>	
<b>Resultado esperado:</b> El sistema debe eliminar de la base de datos los segmentos seleccionados. Se muestra un mensaje de operación satisfactoria.	
<b>Resultado obtenido:</b>	
	
<b>Evaluación de la prueba:</b> Satisfactoria.	

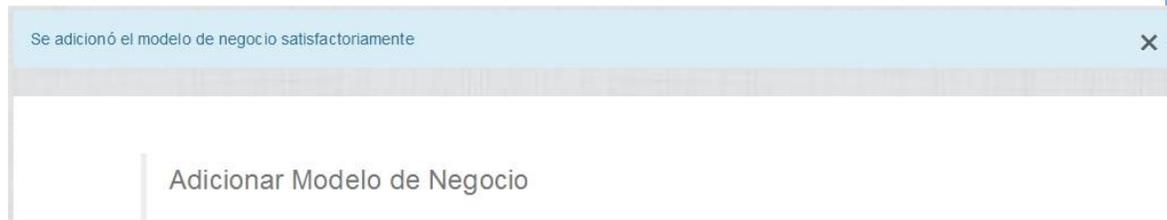
**Tabla 14** Caso de prueba de aceptación para la HU "Insertar Modelo de Negocio".

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> R14_P1	<b>Historia de Usuario:</b> 14
<b>Nombre:</b> Insertar Modelo de Negocio.	
<b>Descripción:</b> El sistema debe permitir al especialista comercial la inserción de nuevos modelos de negocio a la base de datos del mismo.	
<b>Condiciones de Ejecución:</b> Usuario con permisos de Especialista Comercial debe estar autenticado.	
<b>Entrada/Pasos de ejecución:</b>	

1. Clic en la opción **“Añadir modelo”** del módulo Modelos de Negocio ubicado en el menú lateral izquierdo del sistema.
2. Introducir los datos en los campos (Nombre y Descripción). Juego de datos (**“Desarrollo a la medida 1”**, **“Este modelo es muy bueno, desde 1991.”**).
3. Clic en el botón **“Aceptar”**.

**Resultado esperado:** El modelo debe ser añadido a la base de datos de la aplicación. Se muestra un mensaje de operación satisfactoria.

**Resultado obtenido:**



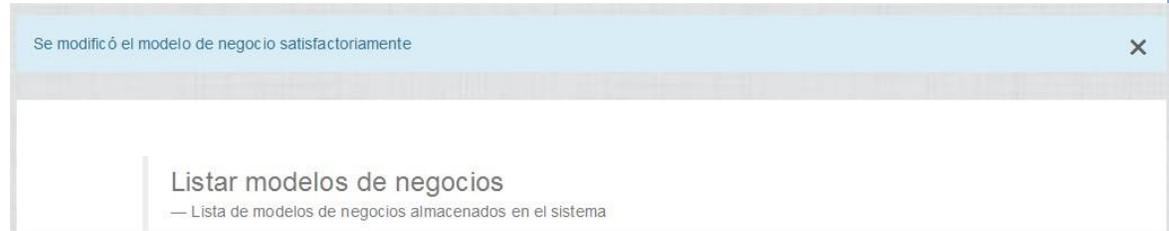
**Evaluación de la prueba:** Satisfactoria.

**Tabla 15** Caso de prueba de aceptación para la HU "Modificar Modelo de Negocio".

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> R15_P1	<b>Historia de Usuario:</b> 15
<b>Nombre:</b> Modificar Modelo de Negocio.	
<b>Descripción:</b> El Especialista comercial puede modificar los valores de los modelos guardados en la base de datos del sistema.	
<b>Condiciones de Ejecución:</b> Usuario con permisos de Especialista Comercial debe estar autenticado y deben existir modelos en la base de datos del sistema.	
<b>Entrada/Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. Seleccionar la opción <b>“Gestionar Modelos”</b> en el módulo Modelos de Negocio que se encuentra en el menú lateral izquierdo.</li> <li>2. Seleccionar el modelo que desea modificar y de este la opción de modificar modelo.</li> <li>3. Introducir los datos del/los campos que se desean modificar (Nombre y Descripción). Juego de datos (<b>“Desarrollo a la medida 2”</b>, <b>“Este modelo es muy bueno, desde 1993.”</b>).</li> <li>4. Clic en el botón <b>“Aceptar”</b>.</li> </ol>	

**Resultado esperado:** El sistema debe modificar en la base de datos los valores introducidos en el modelo. Se muestra un mensaje de operación satisfactoria.

**Resultado obtenido:**



**Evaluación de la prueba:** Satisfactoria.

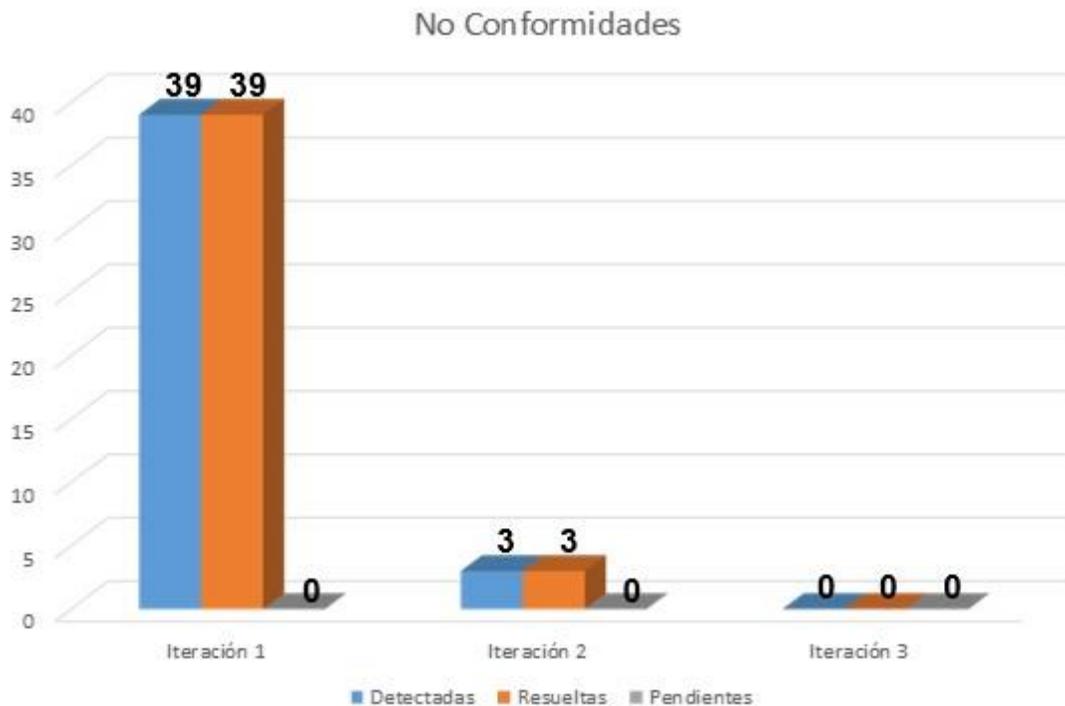
**Tabla 16** Caso de prueba de aceptación para la HU "Eliminar Modelo de Negocio".

Caso de Prueba de Aceptación	
<b>Código:</b> R16_P1	<b>Historia de Usuario:</b> 16
<b>Nombre:</b> Eliminar Modelo de Negocio.	
<b>Descripción:</b> El Especialista comercial puede eliminar los modelos existentes en la base de datos del sistema.	
<b>Condiciones de Ejecución:</b> Usuario con permisos de Especialista Comercial debe estar autenticado y deben existir modelos en la base de datos del sistema.	
<b>Entrada/Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. Seleccionar la opción “<b>Gestionar Modelos</b>” en el módulo Modelos de Negocio que se encuentra en el menú lateral izquierdo.</li> <li>2. Seleccionar el modelo que desea eliminar.</li> <li>3. Clic en la opción “<b>Eliminar</b>”.</li> </ol>	
<b>Resultado esperado:</b> El sistema debe eliminar de la base de datos los modelos seleccionados. Se muestra un mensaje de operación satisfactoria.	
<b>Resultado obtenido:</b>	
<p>The screenshot shows a light blue notification box at the top with the text 'Se eliminó el modelo de negocio satisfactoriamente' and a close button (X). Below the notification, the main content area shows a button labeled 'Listar modelos de negocios' with a subtitle '— Lista de modelos de negocios almacenados en el sistema'.</p>	
<b>Evaluación de la prueba:</b> Satisfactoria.	

### 3.2.2 Resultados de las pruebas

Para la validación de los requisitos funcionales se realizaron 3 iteraciones, realizando en cada una de ellas una prueba de regresión. En la Figura 13 se muestran los resultados obtenidos en cada una de las iteraciones de pruebas realizadas a la plataforma SESMO. Como se puede apreciar en la primera iteración fueron detectadas 39 no conformidades (NC) de las cuales fueron resueltas las 39. Luego en la segunda iteración se encontraron 3 nuevas NC siendo resueltas las mismas. Por último se realizó una tercera iteración, la cual no arrojó nuevas NC. Entre las no conformidades detectadas durante el proceso de pruebas se destacan las siguientes:

- ✓ Los mensajes y botones presentan problemas ortográficos y de idioma.
- ✓ Acciones de eliminar sin confirmación de las mismas.
- ✓ Errores en los formatos de las páginas.



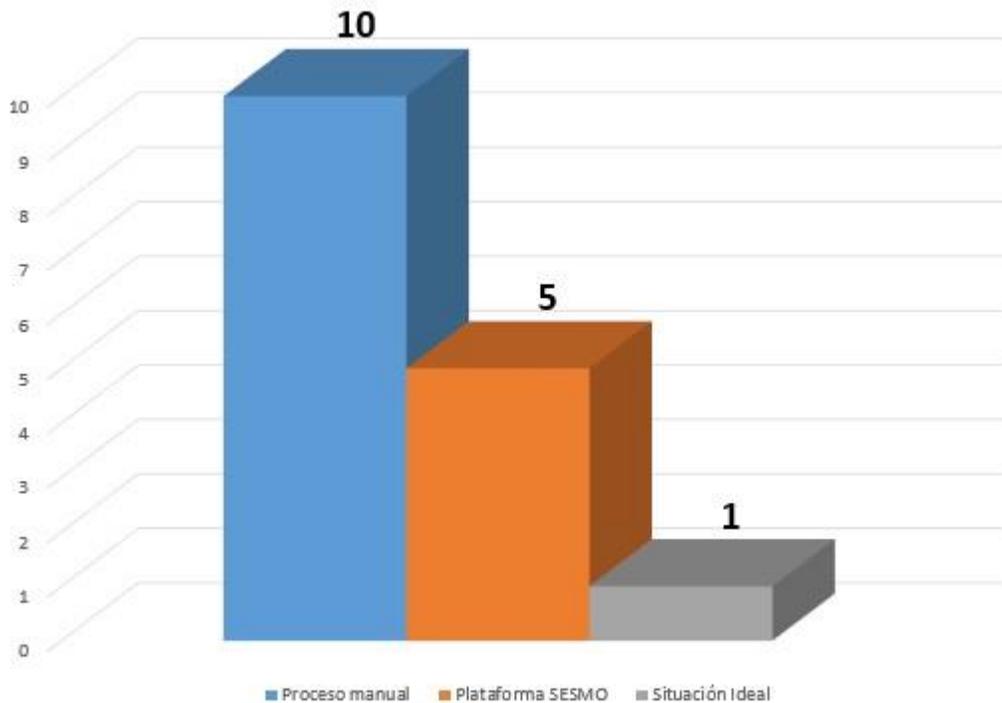
**Figura 11** Comportamiento de las no conformidades por iteración.

Las pruebas de aceptación permitieron demostrar que el sistema cumple con todas las exigencias del cliente, posibilitando así la entrega a este de un producto con un alto nivel de satisfacción por parte del mismo.

### 3.2.3 Comparación de la plataforma desarrollada con el proceso manual.

Actualmente el GIE del CEDIN realiza el proceso de selección de los modelos de negocio para la comercialización de los productos, de forma manual, el mismo demora un período de alrededor de diez días. Ya que los especialistas deben buscar toda la información necesaria, para luego procesarla y seleccionar el modelo. Teniendo en cuenta que la información referente a los modelos de negocio que emplea la competencia muchas veces es muy difícil de encontrar, se tiene un promedio de cinco días para esta actividad y el resto del tiempo para realizar un análisis exhaustivo de todos los elementos encontrados y seleccionar el modelo. El uso de la plataforma SESMO permitirá a los especialistas comerciales realizar este proceso en un promedio de cinco días, lo cual pudo ser comprobado durante el período de pruebas realizadas al sistema, por los especialistas.

El proceso se logra agilizar, pues la herramienta realiza de forma automática la selección del modelo de negocio. En el mejor de los casos (cuando toda la información necesaria ya se encuentra en la base de datos del sistema) el proceso solo tomaría para su realización un día.



**Figura 12** Valoración de la plataforma SESMO teniendo en cuenta el tiempo.

Teniendo en cuenta estos datos se puede afirmar que con el desarrollo del sistema se cumplió con el objetivo general de la investigación, dándole así solución al problema planteado.

### **3.3 Conclusiones Parciales**

Como parte del desarrollo del presente capítulo se determinan las siguientes conclusiones:

- ✓ El uso de los diagramas de componentes y el de despliegue, permitió observar la estructura general del sistema y de la topología de *hardware* donde se ejecuta el mismo.
- ✓ La utilización de los estándares de codificación, permitió optar una estructura homogénea que facilita la comunicación y asegura la calidad, menor cantidad de errores y fácil mantenimiento.
- ✓ La validación del funcionamiento del sistema con la aplicación de las pruebas de aceptación, permitió comprobar el alto grado de satisfacción del cliente con respecto al sistema desarrollado.

## Conclusiones Generales

Al completar el desarrollo de la aplicación *web* se le dio cumplimiento al objetivo planteado inicialmente, obteniendo como resultado principal el sistema SESMO. Con el cual se logró:

- ✓ El estudio de los modelos de negocio empleados para la comercialización de *software*, permitió identificar cuáles son los más usados a nivel mundial y cuáles son los más empleados por el CEDIN.
- ✓ El uso de un conjunto de patrones y buenas prácticas de programación, permitió diseñar e implementar diferentes clases que dan cumplimiento a los requisitos funcionales y no funcionales de la propuesta de solución.
- ✓ La utilización de la metodología AUP-UCI, facilitó la implementación de la plataforma SESMO mediante los artefactos generados.
- ✓ Las pruebas realizadas al sistema permitieron obtener una propuesta que satisface los requisitos funcionales definidos durante la investigación.
- ✓ La utilización de la plataforma SESMO aporta facilidades a los especialistas comerciales en la selección del modelo de negocio más adecuado para la comercialización de los productos del centro, ya que esta agiliza dicho proceso.

## Recomendaciones

Una vez terminada la aplicación y analizados los resultados obtenidos de la presente investigación, se mantiene la idea de continuar perfeccionando la solución final, por lo que se recomienda:

1. Para una futura versión de la aplicación tener en cuenta la importancia de los criterios de medidas, o sea ponderarlos para mejorar la certeza del modelo de negocio seleccionado durante el proceso.
2. Extender el sistema a todos los centros productivos de la Universidad de las Ciencias Informáticas con el fin de agilizar este proceso en cada uno de ellos.
3. Desarrollar una interfaz gráfica para la plataforma teniendo en cuenta la estrategia marcaría de la Universidad.

## Referencias

1. **Definición.de.** [En línea] [Citado el: 3 de Noviembre de 2015.] <http://definicion.de/modelo-de-negocio/>.
2. **Los modelos de negocio y su importancia para la industria del software Libre.** Pascual Riverón, Yadier, y otros. 4, La Habana : Revista Serie Científica, Ediciones Futuro, 2010, Vol. III. ISSN 2306- 2495.
3. **GUTIÉRREZ, J. J.** Qué es un framework web? [En línea] 2011. [Citado el: 16 de Noviembre de 2015.] [www.lsi.us.es/javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/javierj/investigacion_ficheros/Framework.pdf).
4. **SANCHEZ, A. F.** ¿Que es Bootstrap? [En línea] 2013. [Citado el: 16 de Noviembre de 2015.] <http://openwebcms.es/2013/que-es-bootstrap/>.
5. **CUERVO, V.** Hola Mundo con Bootstrap. [En línea] 2013. [Citado el: 16 de Noviembre de 2015.] <http://lineadecodigo.com/bootstrap/hola-mundo-con-bootstrap/>.
6. **What is Symfony?** [En línea] [Citado el: 20 de Noviembre de 2015.] <https://symfony.com/what-is-symfony>.
7. **About Yii.** [En línea] [Citado el: 20 de Noviembre de 2015.] <http://www.yiiframework.com/about/>.
8. **MORA, S. L. RUA.** *Programación de aplicaciones web: historia, principios básicos y clientes web.* [En línea] 2002. [Citado el: 15 de Noviembre de 2015.] [ttp://rua.ua.es/dspace/handle/10045/16995..](http://rua.ua.es/dspace/handle/10045/16995..)
9. **SANTOS, H. H.** Maestros del web. *¿Qué es el PHP?* [En línea] 2001. [Citado el: 15 de Noviembre de 2015.] [http://www.maestrosdelweb.com/phpintro/..](http://www.maestrosdelweb.com/phpintro/)
10. **EGUÍLUZ, J.** Libros Web. *Introducción a CSS.* [En línea] 2009. [Citado el: 15 de Noviembre de 2015.] <http://librosweb.es/css/index.html> .
11. **ZAKAS, N. C.** Professional JavaScript for Web Developers. *Professional JavaScript for Web Developers.* s.l. : Indianápolis, D. E. Wiley Publishing, 2009.
12. **FOUNDATION, J.** jQuery. *jQuery Project.* [En línea] 2014. [Citado el: 18 de Noviembre de 2015.] <https://jquery.org/>.
13. **Utilización de UML.** PERDITA STEVENS, R. P. s.l. : ED, 2007, Vol. II.
14. **ALVAREZ, S.** Desarrollo Web. *Sistemas gestores de bases de datos.* [En línea] 2007. [Citado el: 19 de Noviembre de 2015.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html> .
15. **CASILLAS SANTILLÁN, LUIS A y PÉREZ MORA., ÓSCAR .** *Bases de datos en MySQL.* [En línea] 2014. [Citado el: 19 de Octubre de 2015.] [http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06\\_M2109\\_02151.pdf](http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02151.pdf). ISSN 02151.

16. **MARTINEZ, R.** PostgreSQL. *Sobre PostgreSQL*. [En línea] 2010. [Citado el: 19 de Noviembre de 2015.] [http://www.postgresql.org.es/sobre\\_postgresql..](http://www.postgresql.org.es/sobre_postgresql..)
17. **MORALES, P.** *Servidores Web*. [En línea] 2009. [Citado el: 20 de Noviembre de 2015.] [http://www.monografias.com/trabajos75/servidores-web/servidores-web.shtml. .](http://www.monografias.com/trabajos75/servidores-web/servidores-web.shtml.)
18. **OPENSUSE.** *Apache - openSUSE*. [En línea] 2011. [Citado el: 19 de Noviembre de 2015.] <https://es.opensuse.org/Apache.>
19. **ORACLE.** *Información NetBeans IDE 6.1*. [En línea] 2013. [Citado el: 22 de Noviembre de 2015.] [https://netbeans.org/community/releases/61/index\\_es.html](https://netbeans.org/community/releases/61/index_es.html) .
20. **COMMUNITY, N.** *NetBeans IDE Features*. [En línea] 2013. [Citado el: 20 de Noviembre de 2015.] <https://netbeans.org/features/index.html>.
21. **UML y patrones: introducción al análisis y diseño orientado a objetos.** LARMAN, C. 507, s.l. : Educación, P. PHH, 1999. ISBN 9789701702611.
22. **Postgres.** Introduction. [En línea] 9 de Diciembre de 2013. [Citado el: 20 de Noviembre de 2015.] <http://www.pgadmin.org/index.php>.
23. **Gálvez Lio, Daniel.** Curso de Sistemas Basados en el Conocimiento. *Curso de Sistemas Basados en el Conocimiento*. Villa Clara : s.n., 1998.
24. **ProcesosDeSoftware.** *Metodologías para desarrollo de software*. [En línea] [Citado el: 20 de Octubre de 2015.] <http://procesosdesoftware.wikispaces.com/METODOLOGIAS+PARA+DESARROLLO+DE+SOFTWARE..>
25. **Rodríguez Sánchez, Tamara.** *Metodología de desarrollo para la actividad productiva en la UCI*. La Habana : Publicaciones Futuro, 2014.
26. **Brito, Julio César.** *Módulo Diseñador de Modelos para el Generador Dinámico de Reportes v 2.0*. La Habana : Ediciones Futuro, 2012.
27. **COHN, M.** *User Stories Applied*. 2004. ISBN 0-321-20568-5.
28. **Mestras, P.J.** *Estructuras de las Aplicaciones Orientadas a Objetos. El patrón Modelo-Vista-Controlador (MVC)*. [En línea] 2013. [Citado el: 10 de Febrero de 2016.] <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf> ..
29. **Eguiluz, Javier.** Desarrollo web ágil con Symfony2. 2012.
30. **León, Eduardo.** *Tutorial Visual Paradigm for UML*. [En línea] 2000. [Citado el: 15 de Enero de 2016.] <http://www.slion2000.blogspot.com/>.
31. **Definición de un modelo de datos.** [En línea] 28 de Agosto de 2010. [Citado el: 5 de Febrero de 2016.] <https://tombasededatos.wordpress.com/2010/08/28/2-1-definicion-de-un-modelo-de-datos/> .

- 32. Revisiones de código y estándares de codificación.** *Revisiones de código y estándares de codificación.* [En línea] Microsoft, 2016. [Citado el: 19 de Mayo de 2016.] <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx..>
- 33. PRESSMAN, R.** *Ingeniería de Software.Un enfoque práctico.* España : Mcgraw Hill, 2010. ISBN 970\_10\_5473\_3.
- 35. Estudios de Mercado.** [En línea] 2015. [Citado el: 3 de Noviembre de 2015.] [http://www.estudiosdemercado.org/que\\_es\\_un\\_estudio\\_de\\_mercado.html](http://www.estudiosdemercado.org/que_es_un_estudio_de_mercado.html).
- 36. Definición.de.** [En línea] [Citado el: 3 de Noviembre de 2015.] <http://definicion.de/negocio/>.
- 37. Definición.de.** [En línea] [Citado el: 2015 de Noviembre de 2015.] <http://definicion.de/marketing/>.
- 38. Crowe, Christina.** eHow de España. [En línea] [Citado el: 13 de Enero de 2016.] [http://www.ehowenespanol.com/definicion-estrategia-negocio-sobre\\_377622/](http://www.ehowenespanol.com/definicion-estrategia-negocio-sobre_377622/) .
- 39. Definición.de.** [En línea] [Citado el: 3 de Noviembre de 2015.] <http://definicion.de/toma-de-decisiones/>.

## Glosario de términos

**Estudio de mercado:** “El estudio de mercado es un proceso sistemático de recolección y análisis de datos e información acerca de los clientes, competidores y el mercado. Sus usos incluyen ayudar a crear un plan de negocios, lanzar un nuevo producto o servicio, mejorar productos o servicios existentes y expandirse a nuevos mercados.” (35)

**Negocio:** “El negocio es una operación de cierta complejidad, relacionada con los procesos de producción, distribución y venta de servicios y bienes, con el objetivo de satisfacer las diferentes necesidades de los compradores y beneficiando, a su vez, a los vendedores. En la modernidad, la administración de dichas funciones productivas está a cargo de empresarios y propietarios individuales, quienes a su vez se encargan de organizar y dirigir las industrias, buscando obtener un beneficio económico.” (36)

**Marketing:** “Es un concepto inglés, traducido al castellano como mercadeo o mercadotecnia. Se trata de la disciplina dedicada al análisis del comportamiento de los mercados y de los consumidores. El *marketing* analiza la gestión comercial de las empresas con el objetivo de captar, retener y fidelizar a los clientes a través de la satisfacción de sus necesidades.” (37)

**Comercialización:** No es más que la acción y efecto de comercializar, poner a la venta un producto o darle las condiciones y vías de distribución para su venta.

**Estrategia de negocio:** “es un plan de negocios que toma lugar en el largo plazo para ayudar a alcanzar metas u objetivos específicos. El objetivo de una estrategia de negocios es fortalecer a una empresa particular para que su desempeño mejore, y a cambio, el negocio llega a ser más rentable. Sin una estrategia de negocio, una empresa no tiene una guía para seguir y tiene un riesgo mayor de no ser exitosa.” (38)

**Toma de decisiones:** “Una decisión es una resolución o determinación que se toma respecto a algo. Se conoce como toma de decisiones al proceso que consiste en realizar una elección entre diversas alternativas.” (39)

**Sistema Basado en Conocimiento:** Es un sistema computarizado que usa conocimiento sobre un dominio para arribar a una solución de un problema de ese dominio. Esta solución es esencialmente la misma que la obtenida por una persona experimentada en el dominio del problema cuando se enfrenta al mismo problema. (23)

## Anexos

### Historias de Usuario

Tabla 17 Historia de Usuario "Insertar Producto".

Historia de Usuario	
Número 5	Nombre de la Historia de Usuario: Insertar Producto.
Cantidad de modificaciones a la Historia de Usuario: Ninguna.	
Usuario: Especialista Comercial.	Iteración asignada: 1
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
<p><b>Descripción:</b> El sistema debe permitir al especialista comercial la inserción de nuevos productos a la base de datos del mismo. Para ello el actor involucrado debe de llenar el formulario que el sistema le provee. Dicho formulario cuenta con los siguientes campos: <b>“Nombre”</b>, <b>“Descripción”</b>, <b>“Actividades”</b>, <b>“Precio”</b>, <b>“Fabricante”</b>, <b>“Segmentos”</b>, <b>“Medidas”</b> y <b>“Modelos de Negocio”</b>. Donde el campo nombre se refiere al nombre del producto, este campo solo admite letras (<i>string</i>, 255), el campo descripción se refiere a la descripción del producto y admite valores alfanuméricos (<i>text</i>). Por otra parte el campo fabricante se refiere al nombre del fabricante del producto que puede ser la UCI o cualquier productor de la competencia, este campo es una lista desplegable, la cual muestra todos los fabricantes guardados en la DB del sistema. El campo actividades se refiere a las actividades empleadas durante el desarrollo del producto, este campo será un campo de selección donde se podrá seleccionar cuantas actividades se desee. Los campos segmentos y modelos de negocio se refieren a los segmentos de mercado a los que se encuentra dirigido el producto y los modelos de negocios empleados para su comercialización respectivamente, estos campo serán de selección donde se podrá seleccionar cuantos segmentos y modelos se desee. Además, el campo precio se refiere al precio del producto, este es un campo que solo admite valores numéricos, mientras que el campo medida se refiere a las medidas con las que cumple dicho producto siendo un campo de selección y se podrá seleccionar más de un valor. Una vez llenados dichos campos el actor debe de hacer clic en el botón <b>“Aceptar”</b> y automáticamente el sistema muestra un mensaje de operación satisfactoria. En caso de presentar algún error en el</p>	

contenido introducido por el actor en el formulario, se muestra un mensaje de error en el campo donde se cometió este y se mantiene en dicho formulario.

**Prototipo:**

The screenshot shows a web form titled "Datos del producto" (Product Data). It contains several input fields with associated error messages:

- Nombre del producto:** Input field with error message: "Entre el nombre del producto".
- Descripción del producto:** Text area with error message: "Entre una breve descripción".
- Actividades desarrolladas:** Text area with error message: "— En caso que no encuentre alguna actividad realizada debe añadirla en la base de datos".
- Precio de comercialización:** Input field with error message: "Entre el precio de comercializ...".
- Fabricante:** Input field with value "Wonderware".
- Segmento al que está dirigido:** Input field with error message: "— En caso que no encuentre el segmento debe añadirlo en la base de datos".
- Medida:** Input field.
- Modelo de negocio:** Input field.

An "Aceptar" (Accept) button is located at the bottom right of the form.

**Tabla 18** Historia de Usuario "Modificar Producto".

Historia de Usuario	
Número 6	Nombre de la Historia de Usuario: Modificar Producto.

**Cantidad de modificaciones a la Historia de Usuario:** Ninguna.

**Usuario:** Especialista Comercial.

**Iteración asignada:** 1

**Prioridad en negocio:** Alta

**Riesgo en desarrollo:** Medio

**Descripción:** El sistema debe permitir al especialista comercial la modificación de los atributos de los productos existentes en la base de datos. Esto ocurrirá en caso de que se necesite cambiar la información sobre el producto ya existente. Para ello debe generarse el formulario del producto 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 de operación satisfactoria.

**Prototipo:**

The screenshot shows a web form titled "Datos del producto" (Product Data). The form is organized into several sections:

- Nombre del producto:** A text input field with the placeholder "Entre el nombre del producto".
- Descripción del producto:** A larger text input field with the placeholder "Entre una breve descripción".
- Actividades desarrolladas:** A text input field with a placeholder: "— En caso que no encuentre alguna actividad realizada debe añadirla en la base de datos".
- Precio de comercialización:** A text input field with the placeholder "Entre el precio de comercializ[3]".
- Fabricante:** A dropdown menu with "Wonderware" selected.
- Segmento al que está dirigido:** A text input field with a placeholder: "— En caso que no encuentre el segmento debe añadirlo en la base de datos".
- Medida:** A text input field.
- Modelo de negocio:** A dropdown menu.

At the bottom right of the form is a green button labeled "Aceptar".

**Tabla 19** Historia de Usuario "Eliminar Producto".

Historia de Usuario															
<b>Número 7</b>	<b>Nombre de la Historia de Usuario:</b> Eliminar Producto.														
<b>Cantidad de modificaciones a la Historia de Usuario:</b> Ninguna.															
<b>Usuario:</b> Especialista Comercial.	<b>Iteración asignada:</b> 2														
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio														
<p><b>Descripción:</b> El sistema debe permitir al especialista comercial eliminar productos existentes en la base de datos. Para ello debe de seleccionar el producto a eliminar y ejecutar dicha acción. El sistema debe de mostrar un mensaje de operación satisfactoria.</p>															
<p><b>Prototipo:</b></p> <div style="background-color: #00c853; color: white; padding: 5px; margin-bottom: 5px;">Listado de productos</div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Descripción</th> <th>Precio de comercialización</th> <th>Segmento</th> <th>Medida</th> <th>Fabricante</th> <th>Modelo de Negocio</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>Intouch desarrollar productos y servicios informáticos de automatización industrial, con un alto valor agregado y que cu...</td> <td>195000</td> <td>Industria Eléctrica Industria Petrolera y Gas</td> <td>Gestión de alarmas Generación de reportes Adquisición de datos Registro Histórico de variables Drivers de comunicación</td> <td>Wonderware</td> <td>Modelo Híbrido</td> <td>      </td> </tr> </tbody> </table>		Descripción	Precio de comercialización	Segmento	Medida	Fabricante	Modelo de Negocio	Acciones	Intouch desarrollar productos y servicios informáticos de automatización industrial, con un alto valor agregado y que cu...	195000	Industria Eléctrica Industria Petrolera y Gas	Gestión de alarmas Generación de reportes Adquisición de datos Registro Histórico de variables Drivers de comunicación	Wonderware	Modelo Híbrido	  
Descripción	Precio de comercialización	Segmento	Medida	Fabricante	Modelo de Negocio	Acciones									
Intouch desarrollar productos y servicios informáticos de automatización industrial, con un alto valor agregado y que cu...	195000	Industria Eléctrica Industria Petrolera y Gas	Gestión de alarmas Generación de reportes Adquisición de datos Registro Histórico de variables Drivers de comunicación	Wonderware	Modelo Híbrido	  									

**Tabla 20** Historia de Usuario "Autenticar Usuario".

Historia de Usuario	
<b>Número 1</b>	<b>Nombre de la Historia de Usuario:</b> Autenticar Usuario.
<b>Cantidad de modificaciones a la Historia de Usuario:</b> Ninguna.	
<b>Usuario:</b> Todos.	<b>Iteración asignada:</b> 1
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto

**Descripción:** El sistema debe permitir la autenticación de los usuarios haciendo uso de los servicios LDAP, especificando su usuario y contraseña.

**Prototipo:**



El prototipo muestra un formulario de login con los siguientes elementos:

- Etiqueta "Usuario" sobre un campo de texto con el placeholder "Usuario".
- Etiqueta "Contraseña" sobre un campo de texto con el placeholder "Contraseña".
- Una casilla de verificación con el texto "Recordarme".
- Un botón verde con el texto "Loguearse".

**Tabla 21** Historia de Usuario "Insertar Usuario".

Historia de Usuario	
<b>Número 2</b>	<b>Nombre de la Historia de Usuario:</b> Insertar Usuario.
<b>Cantidad de modificaciones a la Historia de Usuario:</b> Ninguna.	
<b>Usuario:</b> Administrador.	<b>Iteración asignada:</b> 1
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<p><b>Descripción:</b> El sistema debe permitir al administrador del mismo la inserción de nuevos usuarios en la base de datos. Para ello el actor involucrado debe de llenar los campos del formulario “<b>Usuario</b>”, “<b>Correo</b>” y “<b>Rol</b>”, donde el primer campo solo admite palabras y debe de ser un usuario real que se encuentre en la base de datos del LDAP – UCI, mientras que el segundo campo es de tipo “<i>email</i>” y el tercer campo solo admite los valores: “<b>ROLE_USER</b>”, “<b>ROLE_JEFE</b>”, “<b>ROLE_SUBORDINADO</b>” y “<b>ROLE_SUPER_ADMIN</b>”. Una vez llenados dichos campos el actor debe de hacer clic en el botón “<b>Guardar cambios</b>”.</p>	

**Prototipo:**

Crear Usuario

Usuario \*

Correo \*

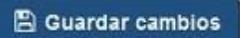
Roles \* 0 \*

✖ Eliminar este elemento

+ Añadir otro elemento

 [Volver al listado](#)

Tabla 22 Historia de Usuario "Modificar Usuario".

Historia de Usuario	
Número 3	Nombre de la Historia de Usuario: Modificar Usuario.
Cantidad de modificaciones a la Historia de Usuario: Ninguna.	
Usuario: Administrador.	Iteración asignada: 1
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
<p><b>Descripción:</b> El sistema debe permitir al administrador modificar los datos de un usuario existente en la base de datos. Esto ocurrirá en caso de que se necesite cambiar el rol del usuario. Para ello debe generarse el formulario del usuario 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 de operación satisfactoria.</p>	
<p><b>Prototipo:</b></p> <p>Usuario * <input type="text"/></p> <p>Correo * <input type="text"/></p> <p>Roles * 0 * <input type="text" value="ROLE_USER"/></p> <p style="text-align: right;">✖ Eliminar este elemento</p> <p style="text-align: right;">+ Añadir otro elemento</p> <p> <a href="#">Volver al listado</a></p>	

**Tabla 23** Historia de Usuario "Eliminar Usuario".

Historia de Usuario															
<b>Número 4</b>	<b>Nombre de la Historia de Usuario:</b> Eliminar Usuario.														
<b>Cantidad de modificaciones a la Historia de Usuario:</b> Ninguna.															
<b>Usuario:</b> Administrador.	<b>Iteración asignada:</b> 2														
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio														
<p><b>Descripción:</b> El sistema debe permitir al administrador eliminar usuarios existentes en la base de datos del mismo. Para ello debe de seleccionar el usuario que desea eliminar de manera permanente del sistema y hacer clic en la opción "<b>Borrar</b>". El sistema debe de mostrar un mensaje de confirmación para dicha operación y el actor debe de hacer clic en el botón "<b>Aceptar</b>".</p>															
<p><b>Prototipo:</b></p>  <table border="1"> <thead> <tr> <th>ID</th> <th>Usuario</th> <th>Correo</th> <th>Roles</th> <th>Habilitado</th> <th>Bloqueado</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>30</td> <td>ymaren</td> <td>ymaren@uci.cu</td> <td>ROLE_SUBORDINADO, ROLE_USER</td> <td>NO</td> <td>SI</td> <td>Ver Modific Borrar</td> </tr> </tbody> </table>		ID	Usuario	Correo	Roles	Habilitado	Bloqueado	Acciones	30	ymaren	ymaren@uci.cu	ROLE_SUBORDINADO, ROLE_USER	NO	SI	Ver Modific Borrar
ID	Usuario	Correo	Roles	Habilitado	Bloqueado	Acciones									
30	ymaren	ymaren@uci.cu	ROLE_SUBORDINADO, ROLE_USER	NO	SI	Ver Modific Borrar									