

INSTITUTO SUPERIOR POLITÉCNICO “JOSÉ ANTONIO ECHEVERRÍA”  
FACULTAD DE INGENIERÍA INDUSTRIAL  
CENTRO DE ESTUDIOS DE INGENIERÍA Y SISTEMA  
INGENIERÍA EN INFORMÁTICA



**SISTEMA INTEGRAL PARA EL SOPORTE DE SOFTWARE**  
**Trabajo para optar por el título de Ingeniería en Informática**

Autor: Melquiades Jorge Mentado

Tutor: Ing. Yariel Quintana Briche

Ciudad de la Habana

Junio de 2005

## ***Declaración de autoría.***

---

Declaro que soy el único autor de este trabajo y autorizo al Centro de Estudios de Ingeniería de Sistemas (CEIS) del Instituto Superior Politécnico José Antonio Echeverría (CUJAE) y a la Dirección de Informatización de la Universidad de las Ciencias Informáticas (UCI) a que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 15 días del mes de julio de 2005.

\_\_\_\_\_

Firma del Autor

\_\_\_\_\_

Firma del Tutor

## ***Opinión del usuario del trabajo de diploma.***

---

El Trabajo de Diploma, titulado Sistema Integral para el Soporte de Software, fue realizado en la Dirección de Informatización de la Universidad de las Ciencias informáticas. Esta Dirección considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface.

- Totalmente
- Parcialmente en un \_\_\_\_ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

---

---

---

---

---

---

---

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a \_\_\_\_\_.

Y para que así conste, se firma la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Representante de la entidad

---

Cargo

---

Firma

cuño



“Ser como la sal y la luz, vale más un hecho que mil palabras.”

*Rabindranath Tagore.*

## ***Agradecimientos.***

---

En este último período, especiales a mi cuñada Anita Hernández Fernández, por su ayuda tan grande, y a los suegros Ana Fernández y Miguel Hernández, por querer y cuidar tanto a mi “Machito” no solo por placer.

En todos los tiempos, a toda mi familia, por quererla y ayudarme tanto.

A los vecinos, Matilde y Juan y sus libros de Matemáticas.

En gran medida a la Facultad de Informática de la Universidad de Camagüey, mi Universidad, y a sus profesores, quienes tanto se esfuerzan por hacernos técnicos y seres humanos cualitativamente superiores.

Muy deseados por mí, a mi maestra Lucía.

A todos los del aula, al alto clérigo, siempre prestos a la cooperación, y especialmente al resto de “La Plevé”; Ioana, Enrique, Adys y Yuni. Han sido muy buenos años juntos.

A la Mayti, aparte, le agradezco por la mensajería.

En fin, a la casualidad de existir donde en primer lugar está “el hombre”, donde se me han dado tantas oportunidades y aseguro que tendré muchas otras, donde la expectativa casi siempre se convierte en realidad.

Siempre faltan.

## ***Dedicatoria.***

---

Alguien que no existe físicamente, pudiera aparecer sola aquí, mi abuela Cuchi.

A la Mama, mi Maro, Maritza Mentado Monagas. Un granito por todo tu sacrificio y amor. A Papi, Melquiades Jorge Pereira y a la hermana más linda que tengo, Kenia Jorge Mentado.

Mi tierno "Machito", Fabio Ernesto Jorge Hernández, verás algún día que a ti también la dedico. Esto es parte del tiempo que te he robado.

A mi tan esperada, incondicional y paciente esposa, Tania Hernández Fernández. Quisiera ponerte en todas estas páginas como en todos mis pensamientos te tengo.

A mi abuelo Hernán, siempre transmitiendo "buenos pensamientos", a mis tías, tíos y al resto de mi familia.

## ***Resumen.***

---

El soporte de software se ha desarrollado en los últimos años, debido que muchas empresas han colocado la actividad de soporte en un nivel más importante dentro de su organización.

El soporte de software incluye servicios de consultas sobre los productos comercializados por una empresa. Esto garantiza el seguimiento a las nuevas necesidades que tiene el cliente y a los problemas que se le presentan.

En este trabajo se hace una propuesta de un sistema para la gestión del soporte de software. Para la confección del mismo se utiliza como lenguaje de programación PHP y el gestor de Bases de Datos MySQL. Se utilizó un sistema de manipulación de contenidos (Content Management System) Drupal, el cual brinda las herramientas para la construcción de sitios Web dinámicos. Todas las herramientas utilizadas son de código abierto, las cuales incorporan las ventajas de utilización de este tipo de software.

El sistema permite realizar un seguimiento de incidencias, almacenándose las preguntas realizadas por los clientes, las respuestas y la mensajería entre trabajadores, generada en este proceso.

# Índice.

---

<b>Introducción.....</b>	<b>1</b>
<b>Capítulo 1. Fundamentación Teórica.....</b>	<b>7</b>
<b>1.1. Calidad de Software y Satisfacción del cliente.....</b>	<b>7</b>
<b>1.2. Surgimiento y Evolución del Soporte de Software.....</b>	<b>8</b>
<b>1.3. Herramientas de Soporte de Software.....</b>	<b>9</b>
<b>1.4. Ventajas del Soporte de Software a través del Web.....</b>	<b>10</b>
<b>1.5. Estructura interna del grupo de Soporte de Software.....</b>	<b>11</b>
1.5.1. Estructura Interna del Grupo de Soporte de la UCI.....	12
<b>1.6. Sistemas automatizados existentes para el Soporte de Software.....</b>	<b>14</b>
1.6.1. SOSoft.....	14
1.6.2. Sitio Web de Soporte de Software de Microsoft.....	15
1.6.3. Sitio Web de Soporte de Software de IBM.....	15
<b>1.7. Beneficios esperados de la Propuesta.....</b>	<b>16</b>
<b>1.8. Conclusiones.....</b>	<b>17</b>
<b>Capítulo 2. Herramientas utilizadas en el desarrollo del sistema.....</b>	<b>18</b>
<b>2.1. Tendencias actuales del software libre.....</b>	<b>18</b>
2.1.1. Software Libre.....	18
2.1.2. El Proyecto GNU.....	19
2.1.3. Código Abierto.....	20
2.1.4. Experiencias a nivel mundial.....	21
2.1.5. Experiencias a nivel nacional.....	21
<b>2.2. Lenguaje PHP.....</b>	<b>22</b>
2.2.1. Ventajas de PHP para programar en Web.....	23
<b>2.3. Sistema gestor de Bases de Datos MySQL.....</b>	<b>24</b>
2.3.1. Ventajas de MySQL.....	24
<b>2.4. Sistemas de Gestión de Contenidos (CMS).....</b>	<b>25</b>
2.4.1. Necesidad de los CMS.....	26
2.4.2. CMS comerciales y de código abierto.....	28
2.4.3. Surgimiento y Evolución de los CMS.....	29
2.4.4. Tendencias de los CMS.....	30
2.4.5. Criterios de selección.....	30
2.4.6. Sistema Manipulador de Contenidos: Drupal.....	31
<b>2.5. Metodología RUP. (Proceso Unificado del Rational).....</b>	<b>33</b>
2.5.1. Lenguaje de Modelación Unificado (UML).....	34
2.5.2. Herramienta Rational Rose.....	34
<b>2.6. Conclusiones.....</b>	<b>35</b>
<b>Capítulo 3. Descripción de la solución propuesta.....</b>	<b>36</b>
<b>3.1. Reglas del negocio.....</b>	<b>36</b>
<b>3.2. Descripción de los procesos del negocio propuestos.....</b>	<b>37</b>
3.2.1. Identificación y justificación de actores.....	37

---

3.2.2. Diagrama de casos de uso del negocio. ....	37
3.2.3. Caso de Uso del Modelo de negocio: Solicitar Servicio de Soporte. ....	37
<b>3.3. Diagrama de clases del modelo de objetos. ....</b>	<b>38</b>
<b>3.4. Requerimientos funcionales. ....</b>	<b>38</b>
<b>3.5. Requerimientos no funcionales. ....</b>	<b>40</b>
3.5.1. Requerimientos de Apariencia o interfaz externa. ....	40
3.5.2. Requerimientos de Usabilidad. ....	41
3.5.3. Requerimientos de Rendimiento. ....	41
3.5.4. Requerimientos de Soporte. ....	41
3.5.5. Requerimientos de Portabilidad. ....	41
3.5.6. Seguridad. ....	42
3.5.7. Legales. ....	42
3.5.8. Confiabilidad. ....	43
3.5.9. Ayuda y documentación en línea. ....	43
3.5.10. Requerimientos de Software. ....	43
3.5.11. Requerimientos de Hardware. ....	43
3.5.12. Restricciones en el diseño y la implementación. ....	43
<b>3.6. Descripción del sistema propuesto. ....</b>	<b>44</b>
3.6.1. Concepción general del sistema. ....	44
3.6.2. Definición de los actores del sistema a automatizar: ....	44
3.6.3. Identificación y descripción de los casos de Uso. ....	47
3.6.4. Diagrama de Casos de Uso del Sistema. ....	56
<b>3.7. Conclusiones. ....</b>	<b>57</b>
<b>Capítulo 4. Construcción de la solución propuesta. ....</b>	<b>58</b>
<b>4.1. Diagramas de clases. ....</b>	<b>58</b>
4.1.1. Paquete Drupal. ....	60
<b>4.2. Diseño de la Base de Datos. ....</b>	<b>64</b>
4.2.1. Diagrama de clases Persistentes. ....	64
4.2.2. Modelo de datos (ver anexo 3). ....	65
<b>4.3. Principios de diseño. ....</b>	<b>65</b>
4.3.1. Estándares en la interfaz de la aplicación. ....	66
4.3.2. Concepción general de la ayuda. ....	66
4.3.3. Tratamiento de excepciones. ....	67
<b>4.4. Estándares de codificación. ....</b>	<b>67</b>
<b>4.5. Diagrama de despliegue. ....</b>	<b>68</b>
<b>4.6. Diagramas de componentes. ....</b>	<b>69</b>
4.6.1. Diagrama de componentes del sistema. ....	69
<b>4.7. Conclusiones. ....</b>	<b>72</b>
<b>Capítulo 5. Estudio de factibilidad. ....</b>	<b>73</b>
<b>5.1. Planificación. ....</b>	<b>73</b>
<b>5.2. Costos. ....</b>	<b>76</b>
<b>5.3. Beneficios tangibles e intangibles. ....</b>	<b>77</b>
<b>5.4. Análisis de costos y beneficios. ....</b>	<b>78</b>
<b>5.5. Conclusiones. ....</b>	<b>78</b>
<b>Conclusiones. ....</b>	<b>79</b>

---

<b>Recomendaciones.....</b>	<b>81</b>
<b>Bibliografía.....</b>	<b>82</b>
<b>Glosario de términos.....</b>	<b>84</b>
<b>Anexos: .....</b>	<b>86</b>
<b>Anexo 1. Estructura global del grupo de soporte de la UCI.....</b>	<b>86</b>
<b>Anexo 2. Flujo de trabajo del grupo de soporte de la UCI.....</b>	<b>87</b>
<b>Anexo 3. Tablas del modelo de datos. ....</b>	<b>88</b>

## ***Introducción.***

---

La producción de software se ha incrementado en los últimos años, entre otros factores, debido al vertiginoso desarrollo de la tecnología. Existen empresas de un alto desarrollo y muchas otras pequeñas empresas que se dedican a la producción de software, pero en todas ellas es un punto principal el garantizar la calidad del software y la satisfacción de los clientes que utilizan sus productos.

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del mismo que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que se eleva la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

En general un producto de software con calidad responde ampliamente a las necesidades del cliente y este muestra un alto grado de satisfacción.

Ahora bien, una vez que el producto es vendido al cliente, es necesario darle un seguimiento adecuado, debido a posibles errores o defectos que se puedan presentar y es necesario responder a las inquietudes del cliente. Es por esta razón que surge la necesidad de implantar los servicios de soporte de software.

El soporte de software abarca todas las actividades que están vinculadas a la manipulación de un software una vez adquirido, así mismo se relaciona con la capacidad del software de satisfacer los requerimientos para su ejecución y funcionalidad durante su vida operacional.

Este servicio se encuentra desarrollándose cada vez más. Las empresas productoras de software han tomado conciencia de la necesidad de su implantación, considerándolo una función tan importante como la propia producción y en la mayoría se ha creado un grupo de soporte de software donde se reúnen especialistas capaces de atender las inquietudes del cliente, utilizando para ello distintas herramientas. Por

este motivo se plantea que inicialmente el soporte de software comenzó de forma desordenada y que actualmente ha llegado a convertirse en una profesión a pesar de mantenerse los métodos iniciales de soporte de software, o sea, la presencia física del grupo de soporte ante el cliente que lo solicita y la utilización del teléfono como vía para comunicar la necesidad de soporte; debido al surgimiento y desarrollo de Internet y de las redes en general, se han implantado como herramientas principales las basadas en la Web. Es común encontrar para la mayoría de las empresas e instituciones productoras de software, en mayor o menor escala, un sitio Web de Soporte de Software.

A nivel internacional, en empresas e instituciones productoras de software, la opción básica que se encuentra con frecuencia es la de brindar soporte de software *por productos* que se ofrecen. De forma alternativa, se brindan las actualizaciones y parches que surgen a medida que se detectan errores en el funcionamiento de cada uno de los productos desarrollados. Las FAQ están presentes en casi la totalidad de los sitios visitados, así como, la posibilidad de contactar con especialistas para que brinden un soporte asistido específico a las necesidades del usuario.

En los sitios más actualizados, aparece la posibilidad de realizar consultas a una base de conocimientos, donde se muestran artículos relacionados con los productos y la actividad de soporte en general, el empleo de foros de discusión y la publicación de noticias referidas precisamente al soporte de software que se brinda.

En las universidades de Latinoamérica no es común encontrar sitios de soporte de software. En otras hispano hablantes el soporte de software se brinda de forma básica, un ejemplo es la UAM en España que brinda soporte de software a determinados productos conveniados con empresas productoras como Microsoft y a sus propios productos. En el caso de las universidades norteamericanas se brinda soporte de software en cuanto a descarga de productos, parches y actualizaciones y se pueden realizar consultas a grupos de expertos en los temas de soporte de software. En el caso de la Universidad de Massachussets, para citar un ejemplo, se brindan los servicios de soporte in situ, con un precio por hora de consulta.

Para el caso de nuestro país, en grupos de desarrollo de software dentro de

---

empresas como Softel, Sifer, Geocuba, Cubatel, etc., no se encuentran herramientas de soporte de software en sus sitios Web.

Así mismo, en las universidades cubanas, tampoco se encuentran servicios de soporte de software en el Web, aunque debemos considerar que nuestras universidades no constituyen entidades productoras de software, por lo que la presencia de un grupo de de soporte y de un sitio de soporte de software se debe desarrollar con el avance tecnológico y el incremento de la utilización de softwares vinculados al funcionamiento de las universidades, tanto desde el punto de vista docente como administrativo.

Sin embargo, en nuestro país se han realizado estudios para crear una herramienta de soporte de software, nombrada *SOSoft*, que pretende crear una plataforma general para el soporte. Dicha herramienta constituye un avance en el reconocimiento del soporte de software como una necesidad para empresas e instituciones cubanas.

La Universidad de las Ciencias Informáticas (UCI), como todas las universidades cubanas, tiene entre sus objetivos la formación de nuevos profesionales y la superación postgraduada. Pero, esta institución añade una característica especial, que es precisamente la producción de software, lo que provoca que tenga en paralelo a la docencia un funcionamiento similar a una entidad productora de software.

En la Dirección de Informatización existen al menos diez sistemas informáticos identificados como prioritarios en el funcionamiento de la Universidad y otros que resuelven un conjunto de problemas de menor impacto.

Actualmente, cuando se implementa un producto y se entrega a un área específica, solo queda la conexión con los propios desarrolladores para viabilizar situaciones de las más diversas índoles, creándose así una dependencia total del equipo de desarrollo, y en ocasiones, de una sola persona en particular.

Este fenómeno priva a la entidad de ofrecer soluciones con los niveles de aceptación requeridos y con la mayor brevedad, creando inconformidad entre los clientes y usuarios finales de las aplicaciones.

Además al ser los propios desarrolladores quienes atienden directamente al cliente,

se atenta contra el tiempo que pudieran emplear en el desarrollo de un nuevo producto.

Desde el surgimiento de la UCI, se ha creado un Grupo de Soporte de Software, con el objetivo de brindar soporte a los productos que se utilizan. Esta estructura existe, pero no ha sido posible ofrecer servicios debido a que, no está organizado el proceso de soporte desde la llegada de la pregunta hasta la búsqueda y ofrecimiento de la respuesta, el personal del grupo de soporte no cuenta con la preparación ni experiencia acumulada en los productos que le permitan ofrecer soluciones y por último no existe una herramienta que brinde servicios en la intranet para consultas rápidas y de menor impacto. Todo esto influye negativamente en los procesos de respuesta al cliente y en la toma de decisiones.

O sea, que actualmente, la situación real que se presenta es que son los equipos de desarrollo los que brindan servicios de soporte a sus propios sistemas, siendo el único personal capacitado para brindar soluciones de cualquier tipo relacionadas con sus aplicaciones. Cuando se recibe alguna solicitud de soporte, no existe otro personal preparado técnicamente ni con experiencia en la solución de un determinado grupo de problemas.

Existe por lo tanto la necesidad de concentrar los esfuerzos de los equipos de desarrollo solo en la evolución de los sistemas, creando un agente intermedio que pueda interactuar directamente con los usuarios y resolver de forma rápida las inquietudes planteadas por ellos, reutilizando soluciones que fueron dadas con anterioridad a los problemas y remitiéndose al equipo de desarrollo solo en casos imprescindibles, previa evaluación.

Por lo anteriormente analizado, se puede plantear que el **objeto de estudio** se enmarca en la calidad de software y el **campo de acción** es el Soporte de Software.

Tomando en consideración las condiciones en que se resuelve actualmente el soporte de software en la Dirección de Informatización de la Universidad de las Ciencias Informáticas, el **problema** del presente trabajo, se formula de la siguiente forma: ¿Cuenta el grupo de Soporte de la Dirección de Informatización de la Universidad de las Ciencias Informáticas con una herramienta para apoyar la labor de Soporte de Software a sus clientes?

---

El **Objetivo general** es desarrollar un sistema integral para el soporte de software para el Grupo de Soporte de Software de la Dirección de Informatización de la Universidad de las Ciencias Informáticas.

Los objetivos específicos son:

- Definir la organización del grupo de soporte de software en niveles de responsabilidad.
- Proporcionar una herramienta de apoyo al trabajo de los integrantes del grupo de soporte.
- Permitir el almacenamiento de las preguntas realizadas por los clientes. Así como su seguimiento, referido a cuáles miembros del grupo de soporte se les ha asignado una pregunta y la mensajería que se genera.
- Creación de una base de conocimientos acerca de la relación problema-solución para cada uno de los productos de la Dirección.
- Permitir el acceso a los usuarios sin necesidad de autenticarse a las FAQ y a la lectura del desarrollo de los temas que han sido propuestos con anterioridad en los foros de discusión.

Las tareas serían las siguientes:

- Realizar entrevistas al personal de la Dirección de Informatización y su Grupo de Soporte para el diagnóstico de la situación real existente.
- Realizar una búsqueda bibliográfica sobre los métodos para brindar soporte de software. Detallar las opciones que generalmente aparecen en sitios Web de soporte de software.
- Realizar un estudio acerca de las herramientas para construir sitios Web.
- Seleccionar cual herramienta se va a utilizar dependiendo de las características del sitio que se desee construir.

- Elaborar los fundamentos teóricos
- Profundizar en la herramienta seleccionada a través de libros y manuales en Internet.
- Confeccionar el modelo del negocio.
- Hacer la lista de requerimientos funcionales y no funcionales del sistema.
- Realizar y documentar el análisis del sistema
- Realizar un estudio de factibilidad del sistema.
- Diseñar la interfaz de usuario

El documento de tesis está estructurado en 5 capítulos. En el capítulo 1 se ofrece una introducción al tema de soporte de software, se abordan los sistemas de soporte que existen a nivel nacional e internacional y se compara la propuesta con otros sistemas existentes. En el capítulo 2 se describen las tecnologías utilizadas en el desarrollo del sistema, en general una fundamentación de las herramientas utilizadas. En el capítulo 3 se realiza la descripción del sistema propuesto. El capítulo 4 se refiere a la construcción de la solución propuesta y por último en el capítulo 5 se realiza un estudio de la factibilidad del sistema.

## **Capítulo 1. Fundamentación Teórica.**

---

En este capítulo se abordan los conceptos fundamentales relacionados en el desarrollo del trabajo. Se comienza con los referidos a la calidad y soporte de software. Así mismo se nombran algunas herramientas para brindar soporte y se especifican las características de las basadas en el Web. Además se explican los modelos de organización de los grupos de soporte y se especifican para el caso de la Dirección de Informatización de la Universidad de las Ciencias Informáticas. Se nombran otros sistemas para el soporte de software, se detallan sus características y se especifican las diferencias con la propuesta.

### **1.1. Calidad de Software y Satisfacción del cliente.**

La calidad es un tema que ocupa a todas las empresas productoras de software. Para lograr que los productos sean utilizados se requiere retener la atención del cliente potencial, es decir el producto o el servicio debe poseer características que coincidan con los requerimientos del cliente, requerimientos que en última instancia son la representación de sus necesidades y expectativas [GON97].

A diferencia de las características del producto o servicio que están bajo el control de una entidad productora, los "requisitos" dependen fundamentalmente del cliente y son la concreción o representación de sus necesidades y expectativas. Por lo que se tiene, de un lado, al cliente con sus necesidades y expectativas, es decir con sus "requisitos" y por el otro, a la entidad productora con unos productos o servicios con unas determinadas características. Entonces, el grado en que las características inherentes de un producto o servicio "cumplen" con unos requisitos (necesidades y expectativas) del cliente es lo que se conoce como Calidad.

Calidad en un software, es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad[FER95].

La norma ISO 9000 plantea textualmente la siguiente definición de calidad: "Grado en que un conjunto de características inherentes cumple con unos requisitos". La frase "Conjunto de características inherentes" se refiere a que todo producto o servicio posee un conjunto de características que le son inherentes, las cuales lo hace diferente y el término "requisitos", la misma norma lo define como "Necesidad o Expectativa establecida generalmente implícita u obligatoria."

La calidad es algo con lo que las organizaciones trabajan a diario, si se elaboran productos con unas características que no tienen nada que ver con los requisitos del cliente o sea, sin tener en cuenta las necesidades y expectativas de este, estarán muy lejos de ser productos de calidad, pero si por el contrario, se toma como base para el diseño y fabricación del producto los requisitos del cliente, los cuales debe conocer de antemano, estará fabricando productos cada vez de mejor calidad [CRU97].

La satisfacción de las necesidades del cliente determina en gran medida la percepción por el cliente de la calidad. Actualmente, las empresas luego de vender sus productos deben brindar seguimiento a las nuevas necesidades que tenga el cliente, por lo que se brinda el servicio de consulta o soporte sobre los productos comercializados como solución a este problema [ARZ02].

## ***1.2. Surgimiento y Evolución del Soporte de Software.***

El soporte de software abarca todas las actividades que están vinculadas a la manipulación de un software una vez adquirido, así mismo se relaciona con la capacidad del software de satisfacer los requerimientos para su ejecución y funcionalidad durante su vida operacional. Así mismo soporte de software se relaciona con el punto de contacto para clientes que buscan asistencia [CON05].

En la década del 90 el mercado demandaba principalmente soporte a hardware, no existía gran conocimiento sobre el manejo y uso de las computadoras personales. Posteriormente comenzó a obtenerse conocimiento y uso de las herramientas de cómputo y a necesitarse también el soporte al software. Los clientes se desplazaban hasta las instalaciones de los vendedores de equipos o software en busca de la solución a sus inquietudes.

En 1992 comienza el soporte a corporativos, con los clientes desde sus puestos de trabajo (On Site), lo cual permitió que se elevara la calidad de los productos. El personal que brindaba el soporte se encontraba dividido; unos en su propia empresa y otros en las instalaciones de los clientes.

En 1996, se comienza a brindar soporte por vía telefónica, pero en ese tiempo la mayor parte del soporte se hacía On Site. En 1997 comienza a ofrecerse el servicio por vía remota a través de las redes de computadoras, manteniéndose la vía On Site y la telefónica.

Las empresas tendieron a incrementar el soporte por vía telefónica, y comenzó a reducirse el servicio On Site. En el año 2000, la comunicación para brindar soporte entre una empresa y sus clientes era generalmente por teléfono o por vía remota a través de redes de computadoras.

A partir de ese momento, comienza una gran preocupación de las empresas por brindar servicios de Soporte de Software efectivos, por lo que incorporan todos sus esfuerzos en el desarrollo de los servicios de soporte vía remota, desarrollando diferentes herramientas de soporte de software.

### ***1.3. Herramientas de Soporte de Software.***

A continuación se ofrece una descripción de cómo utilizar algunas herramientas existentes en servicios de soporte de software [ARZ02].

Utilización del sistema telefónico: Se utilizan los servicios telefónicos para la comunicación entre el cliente y los miembros del grupo de soporte. De esta forma el cliente comunica su inquietud y adquiere una respuesta.

Bases de Conocimientos: Puede estar apoyado el soporte en las Bases de Conocimientos, en las cuales se pueden hacer Búsquedas de Información con rapidez. Estas pueden estar en línea o en CD.

Soporte In Situ: El grupo de soporte o un miembro del mismo, hace presencia física en los locales del cliente donde se presenta la inquietud, previa comunicación al grupo de la existencia del problema.

FAQ: Los documentos de preguntas mas frecuentes recogen las inquietudes que se les presentan a los clientes con sus correspondientes respuestas. Estos documentos pueden ofrecerse tanto en la Intranet como en Internet, en un CD o en documentación impresa. Estos documentos son muy útiles ya que pueden aclarar las dudas básicas de instalación y funcionamiento de los sistemas.

E-mail: Se puede utilizar el correo electrónico (E-mail), permitiendo la comunicación personal entre todos los usuarios de la red. El correo electrónico es una herramienta útil para diseminar alertas sobre un producto. La desventaja es que es permitido comunicaciones del tipo una a muchas, pero no es fácil sostener discusiones en grupo.

Listas de discusión: Las listas de discusión son muy útiles para distribuir información de soporte, en cuanto a lo que se refiere a actualizaciones y ofertas de nuevos productos. Estas listas deben poseer un moderador que controle los mensajes a publicar.

Foros de Discusión: Pueden desarrollarse a través de Internet o la Intranet, se pueden proponer temas, dudas, inquietudes o sencillamente publicar un comentario el cual genere todo un intercambio de opiniones entre los propios clientes y en el cual pueden participar los miembros del grupo de soporte. Es un medio electrónico informal donde los miembros de la comunidad pueden compartir sus conocimientos y experiencias con otros. Los usuarios encuentran respuestas a sus preguntas examinando las preguntas respondidas previamente enviadas por otros.

#### ***1.4. Ventajas del Soporte de Software a través del Web.***

El hecho de que millones de usuarios a través de Internet, agrupados en cientos de redes, puedan intercambiar ficheros de forma regular y acceder al infinito caudal de información disponible, con un considerable ahorro de tiempo y dinero, es la razón principal por la que se estima que una de las herramientas de Soporte de Software más efectiva es el Sitio Web de Soporte de Software.

Entre las ventajas de brindar el Servicio de Soporte de Software a través de un Web se encuentran: Ahorro de tiempo y dinero tanto para la empresa productora de software como para el cliente; estar cerca del cliente, aunque geográficamente se encuentren

lejos; permite brindar al cliente información adicional, no sólo lo relacionado con sus inquietudes; dar respuesta en menor tiempo a los asuntos más sencillos que desee tratar el cliente; promueve la comunicación entre varios clientes de la empresa mediante foros de discusión; brindar Soporte de Software las 24 horas del día; comodidad para el cliente, ya que cualquier consulta la puede hacer desde su puesto de trabajo.

Todos estos beneficios indican que en la actualidad el mejor modo de brindar el servicio de Soporte de Software es utilizando un sitio Web y puede hacerse a través de Internet o de una Intranet, lo cual no significa que las demás herramientas para brindar este servicio se desprecien, lo que ocurre generalmente es la combinación de estas.

Actualmente los Grupos de Soporte de software desarrollan herramienta son conocidas alternativamente como; Help Desk, HotLine, information center, lab assistance, o sencillamente sitio de soporte [BEI05].

### ***1.5. Estructura interna del grupo de Soporte de Software.***

Un aspecto fundamental dentro de una empresa para brindar el servicio de Soporte de software es definir la estructura que tendrá el grupo de Soporte de Software, para facilitar su trabajo.

Existen diferentes aspectos por los cuales un grupo de soporte se puede organizar, a continuación se detallan los modelos de estructura dependiendo del proceso a realizar ante la llegada de un problema. Para ello existen dos modelos: el modelo Frontline/Backline (FL/BL) y el Modelo Touch and Hold (T&H).

En el Modelo Frontline/Backline (FL/BL) se organiza el centro de soporte en dos grupos: Frontline (FL): un grupo de mayor cantidad de personas de menor experiencia, las cuales toman las solicitudes que ingresan al centro y tratan de resolverlas en un tiempo determinado (corto período de tiempo) y Backline (BL): un grupo de mayor experiencia, integrado por menor cantidad de personas, quienes toman las solicitudes que el primer grupo de la FL no pudo resolver [BEI05]. El modelo FL/BL no es bueno desde el punto de vista del cliente debido, entre otros factores, a que si el cliente conoce que el problema es muy difícil no puede llamar directamente a la BL. Este modelo debe ser aplicado cuando las peticiones no tienen gran complejidad o cuando el

volumen de peticiones no es grande y si se tiene una amplia base de conocimientos que guarde gran cantidad de respuestas a dudas de los clientes. Se utilizará sobre todo en organizaciones que comercializan pocos productos o en las que estos no sean muy complejos [ARZ02].

El Modelo Touch and Hold plantea la distribución en dos grupos al igual que el modelo FL/BL: Grupo menos experimentado y otro Grupo de consultas técnicas (TAG). Cuando llega la solicitud, es asignada mediante un algoritmo a un miembro del primer grupo. Este es responsable de tomar la solicitud y para resolver el asunto, utiliza cualquiera de los medios disponibles, incluyendo su conocimiento, documentación del producto, bases de conocimientos técnicos y en particular usando el TAG en capacidad consultativa. El TAG es responsable de Brindar servicio de Soporte a los miembros del primer grupo cuando este requiere consejos u otras fuentes de información y de crear una infraestructura de referencias materiales acerca de cómo buscar y solucionar problemas con la información disponible. Aunque él no trabaje directamente con el cliente, una vez que se decide que se necesita su ayuda, este participa en las conferencias, reuniones y llamadas al cliente. Es extremadamente raro que un miembro del TAG tome propiedad completamente del asunto con el cliente. En resumen el modelo T&H consta de un pequeño grupo de técnicos altamente expertos quienes funcionan mayormente como Soporte a un grupo mayor de personas menos experimentadas [ARZ02].

### **1.5.1. Estructura Interna del Grupo de Soporte de la UCI.**

El Grupo de Soporte de Software de la Dirección de Informatización de la UCI debe reorganizarse a propuesta del Jefe del Grupo de Soporte en dos niveles, dependiendo de las responsabilidades ante el soporte de software. A estos niveles se les ha llamada Receptores/Generalizadores al primero y Especialistas del Grupo al segundo nivel (ver anexo 1).

Entre las responsabilidades de los miembros del primer Nivel Receptores o Generalizadores se encuentran:

Deben actuar como punto único de entrada para las solicitudes del cliente/usuario.

- Tomar las solicitudes que soliciten servicio o reporten problemas.
- Procesar las solicitudes de cualquier tipo, recibidas por otros medios (Correo electrónico, Jabber, vía telefónica, etc.)
- Registrar con exactitud los problemas y sus soluciones en el sistema de incidencias.
- Ejecutar el análisis y diagnóstico inicial del problema reportado.
- Resolver el mayor número de problemas en el primer contacto, basándose en la base de conocimientos disponible y la experiencia adquirida.
- Canalizar los problemas no resueltos al grupo de segundo nivel.

Entre las responsabilidades de los miembros del 2do Nivel (Especialistas del Grupo) se encuentran:

- Servir como recurso especializado en la resolución del problema.
- Resolver problemas.
- Hacerse responsable del problema.
- Mantener informado al cliente.
- Proporcionar el estado del problema cuando sea requerido.
- Registrar con exactitud los problemas y sus soluciones en la base de conocimientos.
- Trabajar conjuntamente los problemas no resueltos con el proveedor del software.

De esta forma son solo los especialistas del segundo nivel quienes interactúan con los Desarrolladores de los sistemas en situaciones específicas o requeridas, nombrados además trabajadores de Nivel tres (ver anexo 2).

Esta propuesta de organización del grupo de soporte de la UCI responde al modelo Frontline/Backline, en este caso el grupo Frontline está compuesto por los recepcionistas, quienes pueden darle al cliente una respuesta y el grupo Backline, está compuesto por los especialistas, los cuales interactúan con el cliente cuando el recepcionista no puede responder la inquietud de este.

## **1.6. Sistemas automatizados existentes para el Soporte de Software.**

### **1.6.1. SOSoft.**

*SOSoft* es una propuesta de una herramienta para automatizar el servicio de Soporte de Software, que puede ser utilizada por organizaciones que se dediquen a la producción de software [ARZ02].

Se plantea que es de propósito general debido a que es adaptable a las necesidades de la empresa o institución que la utilice. Con esta propuesta los usuarios, además de obtener toda la información sobre los productos y servicios de la empresa, pueden enviar las solicitudes de cambio de los productos que ya tienen comprados e instalados, o solicitar servicios directamente. La empresa productora de software estará mucho más cerca de sus clientes y podrá detectar sus necesidades y satisfacerlas oportunamente pues contará con una Base de Datos que contiene toda la información de estas solicitudes las cuales serán asignadas a los especialistas de manera rápida y controlará el avance del trabajo.

Es una herramienta para brindar opciones básicas que demanda el servicio, contribuyendo a la solución de los principales problemas existentes en la Industria de Software Cubana.

#### **1.6.1.1. Análisis comparativo de este sistema con la propuesta.**

*SOSoft* y la propuesta de este estudio son aplicaciones para el soporte de software. En el caso particular del sistema propuesto ha sido creado específicamente para brindar soporte en la Universidad de las Ciencias Informáticas.

Ambos sistemas coinciden en permitir las funcionalidades de un grupo de soporte de software como son soporte por productos y comunicación entre el cliente y los

miembros del Grupo de Soporte.

La mayor diferencia entre *SOSoft* y la propuesta de este trabajo es que al tener en cuenta que los grupos de Soporte de Software presentan una estructura interna por niveles, la propuesta posibilita un seguimiento de incidencias completo, permitiendo al cliente conocer en qué estado se encuentra la solución de su solicitud en cualquier momento que este lo desee conocer. Asimismo la propuesta permite controlar el trabajo de los miembros del grupo de soporte y su nivel de participación en la solución de los problemas y controlar el nivel de satisfacción de los clientes.

### **1.6.2. Sitio Web de Soporte de Software de Microsoft.**

A continuación se señalan algunas empresas que se destacan por su servicio de soporte en la Web, entre ellas Microsoft, la cual establece un sitio de soporte de software que recoge de forma muy abarcadora las opciones fundamentales para el cliente.

El sitio de Microsoft presenta noticias actualizadas de soporte y opciones de grupos de noticias. Se encuentra disponible información técnica de soporte y herramientas de ayuda sobre productos Microsoft usando la Base de Conocimientos de Microsoft. Está el soporte asistido, en el cual se puede enviar la consulta a un profesional de soporte, además se puede conocer el estado en que se encuentra la respuesta a su pregunta. Están las presentaciones técnicas (webcast) donde se refiere a las próximas actualizaciones o agregados que se le incluirán a determinados softwares. También tienen un contrato de soporte entre el cliente y la empresa. Se ha establecido que el Ciclo de vida del soporte técnico de Microsoft sea de hasta 10 años. Su sitio oficial de soporte se encuentra en <http://support.microsoft.com/>, donde se pueden encontrar todas las opciones descritas anteriormente.

### **1.6.3. Sitio Web de Soporte de Software de IBM.**

El sitio de IBM de soporte de software brinda acceso a las opciones de soporte de los productos de IBM. Es posible buscar en la base de conocimientos de soporte, bajar las actualizaciones y parches y enviar un reporte de un problema vía online. Para acceder a la mayoría de estos recursos de soporte, se necesita ser un cliente registrado.

El sitio tiene retroalimentación y se puede dar una sugerencia, un comentario general, así como aportar algo a la organización estructural del sitio. Posee una encuesta al usuario para medir su nivel de satisfacción. El ciclo de vida de soporte de software de IBM, es de 3 años. En el caso de las búsquedas en la base de conocimientos, se puede seleccionar qué tipos de documentos se desean visualizar y brinda la posibilidad de organizarlos por relevancia o por fecha de creación. Su sitio oficial se encuentra en <http://www.ibm.com/support/us/>, donde se puede localizar la información a la que se ha hecho referencia.

### **1.7. Beneficios esperados de la Propuesta.**

Inicialmente entre los beneficios esperados, tras la implantación formal de un grupo de soporte, cuentan:

- Punto único de contacto. Los usuarios no tienen que buscar quien esté disponible para resolver su problema.
- Mejor ambiente de trabajo. El usuario se siente más confiado al saber que cuenta con el apoyo necesario para el mejor provecho de la nueva tecnología.
- Responsabilidades y funciones mejor definidas. Desaparece la informalidad y la falta de atención a los problemas como sucede con el soporte informal. Mayor productividad.

Y con la herramienta que se propone para la gestión de soporte:

- El equipo de soporte podrá brindar soluciones más rápidas, y constituirá sin dudas una fuente útil para los desarrolladores en la evolución de sus sistemas.
- Se permite la recepción mediante llamadas telefónicas, correo electrónico, mensajería instantánea u otros medios, de las incidencias que interrumpan la normal operatoria del trabajo con los sistemas de la entidad, o que generen consultas, asesoramiento en el funcionamiento y/o utilización de los recursos informáticos disponibles.
- Seguimiento de incidencias para mantener informados a los usuarios del

estado de las mismas. Control de estadísticas de las incidencias por grupos de resolución, tiempos de resolución, usuarios y otros parámetros. Las mismas tienen como objetivo la realización un análisis de la actividad del grupo, conllevando al mejoramiento del servicio y la operabilidad de los usuarios.

### **1.8. Conclusiones.**

En este capítulo se brindaron los conceptos básicos relacionados con el soporte de software. Se detallan las características organizacionales de los grupos de soporte y se especifica el caso de la Dirección de Informatización de la UCI. Se exponen, además, otras propuestas de sistemas de soporte de software.

## **Capítulo 2. Herramientas utilizadas en el desarrollo del sistema.**

---

En este capítulo se realiza una descripción general de las herramientas utilizadas en la construcción del sistema. Se analizan las ventajas de la utilización del software libre, así como las tendencias y experiencias a nivel internacional y nacional. Se describen las características y ventajas del lenguaje de programación PHP. Se hace referencia a los Gestores de Bases de Datos y las ventajas de MySQL. Se explican los conceptos generales de los Manipuladores de contenido (CMS) y se especifica el caso de Drupal. Por último se realizan breves explicaciones de las herramientas y la metodología utilizadas en el análisis y diseño del sistema.

### **2.1. Tendencias actuales del software libre.**

#### **2.1.1. Software Libre.**

El "software libre" es un tipo particular de software que se basa en que el código fuente de un programa debe estar disponible, para luego poder modificarlo, mejorarlo y distribuirlo libremente. Las ventajas y oportunidades que el software libre puede ofrecer comienzan a ser consideradas muy seriamente en todo el mundo.

El movimiento de Software Libre tuvo sus orígenes en laboratorios informáticos del Instituto de Tecnología de Massachussets (MIT), donde en la década del '70, comenzó a trabajar Richard Stallman, quien es considerado como el creador del movimiento. Los programadores de aquella época podían examinar los programas y modificarlos, sin restricciones, debido a que el código fuente se encontraba disponible junto con el programa

Durante los años 80 muchos programadores del MIT abandonaron el Laboratorio de Inteligencia Artificial y firmaron acuerdos comerciales con diversas empresas. El compromiso de mantener en secreto el software durante sus primeras etapas de desarrollo les permitía el trabajar en los mejores proyectos, por otro lado, como la mayoría de los acuerdos de no revelar (NDAs) tienen fecha de expiración, pensaban

que, tarde o temprano, el software habría de volverse de público conocimiento.

La idea de compartir el código fuente empezó a desaparecer a principios de la década del '80, cuando comenzó la revolución de los microordenadores. Las empresas se dieron cuenta de que la mayoría de los usuarios sólo querían softwares que funcionasen, por lo que no necesitaban el código ni las instrucciones que solo podían entender los programadores. Así, las empresas aprendieron rápidamente que podían guardarse el código fuente para ellos y mantener relativamente satisfechos a sus clientes y al mismo tiempo les cerraban las puertas a sus competidores.

Richard Stallman, ha planteado en reiteradas ocasiones que el término "libre" en la denominación de software libre se refiere a libertad, no a precio. La aclaración es útil para los angloparlantes e interesante para todos, ya que, aunque el software libre suele ser gratuito, no tiene por qué serlo. Lo que le define, sin embargo, es que otorga una serie de libertades a los usuarios, entre ellas: libertad de ejecutar el programa para cualquier propósito, libertad para modificar el código, libertad para copiar y libertad para distribuir los programas, generalmente bajo las condiciones de la GPL [Vil01b].

Permite también a los profesionales de la informática y a los desarrolladores de sistemas un libre intercambio de información e investigación sobre cómo funcionan los programas, de compartir conocimientos y experiencias con este tipo de software: consejos, resolver errores, asistencia, de trabajar en proyectos juntos a través de Internet (remotamente), rompiendo las barreras geográficas y permitiendo que un mismo software sea desarrollado en distintas partes del mundo.

### **2.1.2. El Proyecto GNU.**

GNU es un acrónimo recursivo que significa "GNU No es Unix". El proyecto GNU fue iniciado por el estadounidense Richard Stallman con el objetivo de crear un sistema operativo totalmente libre: el sistema GNU [SIS05]., Se anunció públicamente el 27 de septiembre de 1983, en el grupo de noticias *net.unix-wizards*. Al anuncio original, siguieron otros ensayos como el "Manifiesto GNU", donde se establecieron las motivaciones para realizar el proyecto GNU, entre las que se puede destacar la noción de "retornar al espíritu de cooperación que prevaleció en los tiempos iniciales de la

comunidad de usuarios de computadoras" [GNU05].

UNIX es un sistema operativo propietario (es decir, privativo/cerrado) muy popular, porque está basado en una arquitectura que ha demostrado ser técnicamente estable. El sistema GNU fue diseñado para ser totalmente compatible con UNIX. En 1990 el sistema GNU ya tenía un editor de texto llamado Emacs, un exitoso compilador (GCC), y la mayor parte de las bibliotecas fundamentales y las utilidades que componen una típica distribución UNIX. Pero faltaba el componente primario de todo sistema operativo llamado núcleo.

En 1991, Linus Torvalds, un programador finlandés, quien se encontraba para ese entonces estudiando en la Universidad de Helsinki, escribió el núcleo de Linux para hacerlo compatible con UNIX y lo liberó bajo la Licencia Pública General (GPL). Linux fue posteriormente ampliado gracias al trabajo de múltiples programadores a través de Internet. En 1992, el núcleo Linux fue combinado con el sistema GNU, resultando en un sistema operativo libre y completamente funcional. El sistema GNU, que más comúnmente se encuentra de esta forma, es usualmente conocido como "GNU/Linux" o como una "distribución Linux".

Subsecuentemente, miles de programadores voluntarios alrededor del mundo han participado en el proyecto a través de Internet, mejorándolo continuamente. Torvalds y otros desarrolladores de los primeros días de Linux adaptaron los componentes de GNU para trabajar con el núcleo de Linux.

### **2.1.3. Código Abierto.**

Código Abierto (*Open Source*) surgió en 1998 en Palo Alto, California, en una reunión en la que se encontraban entre otros Larry Augustin (de Linux Internacional), y Eric S. Raymond, que es el actual presidente de la OSI (*Open Source Initiative*).

Al adquirir un software de código abierto, se adquiere a la vez el código fuente, de tal forma que se les puede realizar modificaciones según las necesidades de cada programador. Un software de código abierto puede adquirirse gratis o ser software propietario [Vil01a].

El término "Open Source" se refiere a tener acceso al código fuente. Pero el acceso al código fuente es apenas un pre-requisito para dos de las cuatro libertades que definen al Software Libre.

En la actualidad empresas como *Apple*, *Sun* y *Netscape* están experimentando con licencias de código abierto, compartiendo código y tratando de hacer negocios al mismo tiempo. *Sun* incluso comparte el código fuente de su plataforma *Java*.

#### **2.1.4. Experiencias a nivel mundial.**

Entre las experiencias más importantes de empleo de software libre a nivel mundial se encuentran los casos de las ciudades de Bergen en Noruega y Munich en Alemania, la región de Andalucía y algunas prácticas que se están desarrollando en Corea del Sur, Francia, Venezuela y Brasil, sólo para destacar las más relevantes y de mayor impacto [GAR03].

En Venezuela, existe un decreto mediante el cual se establecerá la utilización de software libre para todos los organismos y dependencias de la administración pública. La intención de esta medida es lograr independencia tecnológica e informática.

Por otro lado en Francia, el Ministerio del Equipamiento está trabajando en el proyecto de la migración a Linux, por ello se está sustituyendo la infraestructura de las aplicaciones de oficina de Microsoft y 1.500 servidores Windows NT, por los servidores de Mandrakelinux, Mandrakesoft. Este sistema operativo (distribución de Linux) también fue el elegido para entrenar y ayudar a los usuarios.

También el gobierno de Corea del Sur ha anunciado un plan para migrar miles de computadoras que usan software de Microsoft hacia el software libre. Los ministerios y universidades están incluidos en este plan de migración que esperan poder finalizar en el 2007. Si la migración es exitosa se espera ahorrar 300 millones de dólares por año [GAR03].

#### **2.1.5. Experiencias a nivel nacional.**

Cuba se encuentra trabajando fuertemente en el desarrollo de la Industria Cubana del Software, de manera que supla las necesidades nacionales y se convierta en fuente

de generación de divisas al país.

Esta industria, si bien debe mantener la utilización de plataformas basadas en software propietario, no puede estar ajena a la revolución del software libre, por lo que tiene que estar preparada en esta tecnología, debe crearse la base de desarrolladores de software libre capaz de responder a las demandas nacionales e internacionales en esta esfera.

Las especialidades Universitarias de Informática, los Técnicos Medios, así como el movimiento de los “Joven Club”, en sus planes de estudio, deben profundizar la problemática del software libre y estudiar las distintas herramientas existentes.

Es este aspecto varias instituciones del país han migrado total o parcialmente, sus sistemas a software libre. Esta política del país debe ir en incremento hasta lograr un alto por ciento de usuarios del software libre en el trabajo diario de las empresas e instituciones cubanas [EST02].

En el recién desarrollado III Taller de Software Libre, el cual tuvo lugar durante la Convención Internacional Informática 2005, se conoció la noticia de que Cuba está organizando la migración progresiva de las computadoras instaladas en los organismos de la Administración Central del Estado hacia el software libre, sobre la base del sistema operativo Linux. La noticia trascendió durante una conferencia ofrecida por Roberto del Puerto, director de la Oficina para la Informatización de la Sociedad. La política será rectorada por un Grupo Nacional, que integran entre otros la propia Oficina, los ministerios de Justicia, del Interior, de Informática y las Comunicaciones, la Red Telemática de Salud (Infomed), la CUJAE, los Joven Club y la Universidad de las Ciencias Informáticas [VAL05].

## **2.2. Lenguaje PHP.**

PHP son las siglas de “Hypertext Preprocessor”. Es un lenguaje de programación escrito por Rasmus Lerdorf en 1995, pensado en la Web el cual es ideal para la creación de páginas dinámicas. Sus siglas originalmente significaron: “Personal Home Page” [HIS03].

Una Web dinámica con PHP contiene una serie de documentos PHP que el servidor interpreta proporcionando al cliente documentos HTML con el resultado de las órdenes PHP.

En el desarrollo de aplicaciones Web, Java, como lenguaje símbolo en la programación orientada a objetos, tomó fuerza en los últimos años, pero desde PHP4, con un modelo orientado a objetos bastante limitado, se ofrece lo necesario para programar usando algunas de las ventajas de los objetos. El PHP5 brinda una orientación a objetos bastante parecida al Java, PHP5 presenta mejoras significativas y un ambiente de programación orientada a objetos mucho más completo que permiten que PHP proporcione un alto rendimiento a las aplicaciones Web empresariales a nivel de las plataformas.NET [DON02].

### **2.2.1. Ventajas de PHP para programar en Web.**

Entre las características que influyen en la selección de PHP se encuentra en primer lugar su simplicidad. Su sintaxis está inspirada en el lenguaje C, ligeramente modificada para adaptarla al entorno en el que trabaja, de modo que si se está familiarizado con esta sintaxis, resulta muy fácil aprender PHP.

PHP es suficientemente versátil y potente como para hacer tanto aplicaciones complejas que necesiten acceder a recursos de bajo nivel del sistema como pequeños scripts que envíen por correo electrónico un formulario llenado por un cliente. Viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a Base de Datos, encriptación, envío de correo, gestión de un *e-commerce*, XML creación de PDF...)

El tiempo es uno de los costos más altos que hay que tener en cuenta antes de iniciar un proyecto. Para empezar, el tiempo de aprendizaje de PHP es muy corto debido a su simplicidad. Por lo tanto, el tiempo de desarrollo es también corto. Otro costo que hay que tener en cuenta es el del hardware. Para desarrollar en PHP no se requiere tener grandes capacidades de hardware.

PHP trae funciones especiales para trabajar con el motor de Base de Datos *Open Source* más popular del mercado: *MySQL*. Además existe abundante información,

manuales y tutoriales de *MySQL* con PHP.

PHP es soportado por un gran número de servidores Web, incluyendo el servidor Web *Open Source* más popular: *Apache*. Es muy rápido. Su integración con la Base de Datos *MySQL* y el servidor *Apache*, le permite constituirse como una de las alternativas más atractivas del mercado.

Entre las inconveniencias que se le señaló a PHP fue la no existencia de un IDE o Debugger, hasta que finalmente se propuso un IDE y un debugger nombrado "Zend Studio" [DON02]. Zend Studio como editor hace todo más fácil al completar funciones, acceder a la documentación de las funciones y proveer de una interfaz amigable para el trabajo en PHP.

### **2.3. Sistema gestor de Bases de Datos MySQL.**

La información que puede almacenar una Base de Datos puede ser tan simple como la de una agenda, un contador, o un libro de visitas, o tan compleja como la de una tienda en línea, un sistema de noticias, un portal, o la información generada en una red corporativa. Para agregar, acceder, y procesar los datos almacenados en una Base de Datos, se necesita un sistema de administración de Bases de Datos.

Un Sistema de Gestión de Bases de Datos es un software con capacidad para definir, mantener y utilizar una Base de Datos. Un sistema de gestión de Bases de Datos debe permitir definir estructuras de almacenamiento, acceder a los datos de forma eficiente y segura [SAN02].

*MySQL* es un sistema de gestión de Bases de Datos que fue creado por la empresa sueca *MySQL AB*, que mantiene el derecho de autor del código fuente del servidor SQL. La parte SQL de "*MySQL*" significa "Lenguaje Estructurado de Consulta", y es el lenguaje más usado y estandarizado para acceder a Bases de Datos relacionales

#### **2.3.1. Ventajas de MySQL.**

A continuación se señalan las características de MySQL que llevaron a su selección como gestor de Bases de Datos.

Primeramente, este gestor de Bases de Datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de varios lenguajes de programación, además de su fácil instalación y configuración.

*MySQL* consiste de un sistema cliente/servidor que se compone de un servidor SQL multihilo, varios programas clientes y bibliotecas, herramientas administrativas, y una gran variedad de interfaces de programación (APIs).

*Por último, MySQL es Open Source.* Esto significa que la persona que lo desee puede usar y modificar *MySQL*, o sea, cualquiera que lo necesite puede estudiar el código fuente y cambiarlo de acuerdo a sus necesidades.

## **2.4. Sistemas de Gestión de Contenidos (CMS).**

Realizar un sitio Web puede ser un trabajo complicado y muy laborioso si no se dispone de las herramientas adecuadas. En el pasado, las herramientas eran básicamente editores que permitían generar una página, los cuales evolucionaron para incorporar el control de la estructura de un Web y otras funcionalidades, pero en general estaban enfocados más a la creación que al mantenimiento. En los últimos años se ha desarrollado el concepto de sistema de gestión de contenidos (Content Management Systems o CMS). Se trata de herramientas que permiten crear y mantener un Web con facilidad, encargándose de los trabajos más tediosos que hasta ahora ocupaban el tiempo de los administradores.

Los sistemas gestores de contenidos proporcionan un entorno que posibilita la actualización, mantenimiento y ampliación del Web con la colaboración de múltiples usuarios. Los CMS son softwares que se utilizan principalmente para facilitar la gestión de sitios Web, ya sea en Internet o en una intranet, y por eso también son conocidos como gestores de contenido Web (Web Content Management o WCM).

Hay que tener en cuenta, sin embargo, que la aplicación de los CMS no se limita sólo a las Web. James Robertson [ROB03] propone una división de la funcionalidad de los sistemas de gestión de contenidos en cuatro categorías: creación de contenido, gestión

de contenido, publicación y presentación.

En cuanto a la creación de contenido, un CMS aporta herramientas para que los creadores sin conocimientos técnicos en páginas Web puedan concentrarse en el contenido. Referido a la gestión de contenido, los documentos creados se depositan en una Base de Datos central donde también se guardan el resto de datos del sitio, como son los datos relativos a los documentos, datos y preferencias de los usuarios, la estructura del Web, etc.

En cuanto a la publicación de una página se aplica el patrón definido para toda la Web o para la sección concreta donde está situada, de forma que el resultado final es un sitio Web con un aspecto consistente en todas sus páginas. Esta separación entre contenido y forma permite que se pueda modificar el aspecto visual de un sitio Web sin afectar a los documentos ya creados y libera a los autores de preocuparse por el diseño final de sus páginas.

En cuanto a la presentación, un CMS puede gestionar automáticamente la accesibilidad del Web, con soporte de normas internacionales de accesibilidad y adaptarse a las preferencias o necesidades de cada usuario. También puede proporcionar compatibilidad con los diferentes navegadores disponibles en todas las plataformas (Windows, Linux, Mac, Palm, etc.) y su capacidad de internacionalización le permite adaptarse al idioma, sistema de medidas y cultura del visitante. El sistema se encarga de gestionar muchos otros aspectos como son los menús de navegación o la jerarquía de la página actual dentro del Web, añadiendo enlaces de forma automática. También gestiona todos los módulos, internos o externos, que incorpore al sistema. Así por ejemplo, con un módulo de noticias se presentarían las novedades aparecidas en otro Web, con un módulo de publicidad se mostraría un anuncio o mensaje animado, y con un módulo de foro se podría mostrar, en la página principal, el título de los últimos mensajes recibidos. Todo eso con los enlaces correspondientes y, evidentemente, siguiendo el patrón que los diseñadores hayan creado.

#### **2.4.1. Necesidad de los CMS.**

Se podría pensar que un CMS no es necesario para un sitio Web relativamente

pequeño o cuando no se necesitan tantas funcionalidades. Eso sólo podría ser cierto para un sitio con unas pocas páginas estáticas para el que no se prevea un crecimiento futuro ni muchas actualizaciones, lo que no es muy realista. En cualquier otro caso, la flexibilidad y escalabilidad que permiten estos sistemas, justifican su utilización en prácticamente cualquier tipo de Web.

Estos son algunos de los puntos más importantes que hacen útil y necesaria la utilización de un CMS:

- Inclusión de nuevas funcionalidades en el Web. Esta operación puede implicar la revisión de multitud de páginas y la generación del código que aporta las funcionalidades. Con un CMS eso puede ser tan simple como incluir un módulo realizado por terceros, sin que eso suponga muchos cambios en el Web. El sistema puede crecer y adaptarse a las necesidades futuras
- Mantenimiento de gran cantidad de páginas. En una Web con muchas páginas hace falta un sistema para distribuir los trabajos de creación, edición y mantenimiento con permisos de acceso a las diferentes áreas.
- Páginas interactivas. Las páginas estáticas llegan al usuario exactamente como están almacenadas en el servidor Web. En cambio, las páginas dinámicas no existen en el servidor tal como se reciben en los navegadores, sino que se generan según las peticiones de los usuarios. De esta manera cuando por ejemplo se utiliza un buscador, el sistema genera una página con los resultados que no existían antes de la petición. Para conseguir esta interacción, los CMS conectan con una Base de Datos que hace de repositorio central de todos los datos del Web
- Cambios del aspecto de la Web. Si no hay una buena separación entre contenido y presentación, un cambio de diseño puede comportar la revisión de muchas páginas para su adaptación. Los CMS facilitan los cambios con la utilización, por ejemplo, del estándar CSS (Cascading Style Sheets u hojas de estilo en cascada) con lo que se consigue la independencia de presentación y contenido.

- **Consistencia del Web.** La consistencia en un sitio Web no quiere decir que todas las páginas sean iguales, sino que existe un orden. Un usuario nota enseguida cuándo una página no es igual que el resto de las del mismo Web por su aspecto, la disposición de los objetos o por los cambios en la forma de navegar. Estas diferencias provocan sensación de desorden y dan a entender que el Web no lo han diseñado profesionales. Los CMS pueden aplicar un mismo estilo en todas las páginas y aplicar una misma estructura mediante patrones de páginas
- **Control de acceso.** Controlar el acceso a un sitio Web no consiste simplemente el permitir la entrada al sitio, sino que además se refiere a gestionar los diferentes permisos a cada área, aplicados a grupos o individuos

#### **2.4.2. CMS comerciales y de código abierto.**

Se puede hacer una clasificación de los CMS según el tipo de licencia escogido. Por una parte están los CMS comercializados por empresas que consideran el código fuente un activo más que tienen que mantener en propiedad, y que no permiten que terceros tengan acceso (como es el caso del Share Point de Microsoft). Por la otra tenemos los de código fuente abierto, desarrollados por individuos, grupos o empresas que permiten el acceso libre y la modificación del código fuente.

Los CMS de código abierto son mucho más flexibles, pero se podría considerar que la herramienta comercial será más estable y coherente al estar desarrollada por un mismo grupo. En la práctica esta ventaja no es tan grande, ya que los CMS de código abierto también están coordinados por un único grupo o por empresas, de forma similar a los comerciales [OPE05].

Utilizar una herramienta de gestión de contenidos de código abierto tiene otra ventaja que hace decidirse a la mayoría de usuarios: su costo. Sólo en casos aislados se hacen distinciones en este aspecto entre empresas o entidades sin ánimo de lucro a particulares. En comparación, los productos comerciales pueden llegar a tener un costo que sólo una gran empresa puede asumir.

En cuanto al soporte, los CMS comerciales acostumbran a dar soporte profesional,

con un costo elevado en muchos casos, mientras que los de código abierto se basan más en las comunidades de usuarios que comparten información y solución a los problemas. Las formas de soporte se pueden mezclar, y así encontramos CMS de código abierto con empresas que ofrecen servicios de valor añadido y con activas comunidades de usuarios. En el caso comercial también sucede, pero el costo de las licencias hace que el gran público se decante por otras opciones y por lo tanto las comunidades de soporte son más pequeñas.

Un problema que acostumbra a tener el software de código abierto es la documentación, generalmente escasa, dirigida a usuarios técnicos y frecuentemente mal redactada. Este problema se agrava en el caso de los módulos desarrollados por terceros, que no siempre incorporan las instrucciones de su funcionamiento de forma completa y entendible.

### **2.4.3. Surgimiento y Evolución de los CMS.**

A principios de los años noventa, el concepto de sistemas de gestión de contenidos era desconocido. Algunas de sus funciones se realizaban con aplicaciones independientes: editores de texto y de imágenes, Bases de Datos y programación a medida.

Ya en el año 1994, *Ilustra Information Technology*, utilizaba una Base de Datos de objetos como repositorio de los contenidos de un Web, con el objetivo de poder reutilizar los objetos y ofrecía a los autores un entorno para la creación basado en patrones. La idea no cuajó entre el público y la parte de la empresa enfocada al Web fue comprada por AOL, mientras que *Informix* adquirió la parte de Bases de Datos.

RedDot es una de las empresas pioneras que empezó el desarrollo de un gestor de contenidos en el año 1994. No fue hasta finales del año siguiente que presentaron su CMS basado en una Base de Datos.

Entre los CMS de código abierto uno de los primeros fue Typo 3, que empezó su desarrollo el año 1997, en palabras de su autor, Kasper Skarhoj, "antes de que el término gestión de contenidos fuera conocido sobradamente". PHPNuke, la herramienta que popularizó el uso de estos sistemas para las comunidades de usuarios en Internet,

se empezó a desarrollar el año 2000.

#### **2.4.4. Tendencias de los CMS.**

Sobre el futuro de los CMS, Robertson [ROB03] apunta que los CMS se convertirán en un artículo de consumo cuando los productos se hayan establecido y más soluciones lleguen al mercado. Eso provocará una disminución de los precios en los productos comerciales y una mayor consistencia en las funcionalidades que ofrecen. En este entorno, muchas empresas que implementan sitios Web tendrán que cerrar.

El campo de los gestores de contenido madurará hasta conseguir un alto grado de consistencia y profesionalismo. Se adoptarán estándares en el almacenamiento, estructuración y gestión del contenido. Se producirá una fusión entre gestión de contenidos, gestión de documentos y gestión de registros

#### **2.4.5. Criterios de selección.**

Antes de empezar el proceso de selección de un CMS concreto, hay que tener claros los objetivos del Web, teniendo en cuenta al público destinatario, y estableciendo una serie de requerimientos que tendría que satisfacer el CMS.

La siguiente lista está basada en las funciones principales de los CMS expuestas anteriormente, las indicaciones de Robertson, [ROB02] y una recopilación de los requerimientos básicos de una Web.

- **Arquitectura técnica.** Tiene que ser fiable y permitir la escalabilidad del sistema para adecuarse a futuras necesidades con módulos. También tiene que haber una separación de los conceptos de contenido, presentación y estructura que permita la modificación de uno de ellos sin afectar a los otros. Es recomendable, pues, que se utilicen hojas de estilo (CSS) y patrones de páginas.
- **Grado de desarrollo.** Madurez de la aplicación y disponibilidad de módulos que le añaden funcionalidades
- **Soporte.** La herramienta tiene que tener soporte tanto por parte de los creadores como por otros desarrolladores. De esta manera se puede asegurar de

que en el futuro habrá mejoras de la herramienta y que se podrá encontrar respuesta a los posibles problemas

- Posición en el mercado y opiniones. Una herramienta poco conocida puede ser muy buena, pero hay que asegurarse de que tiene un cierto futuro. También son importantes las opiniones de los usuarios y de los expertos
- Usabilidad. La herramienta tiene que ser fácil de utilizar y aprender. Los usuarios no siempre serán técnicos, por lo tanto hace falta asegurarse que podrán utilizar la herramienta sin muchos esfuerzos y sacarle el máximo rendimiento
- Velocidad de descarga. Teniendo en cuenta que no todos los usuarios disponen de líneas de alta velocidad, las páginas se tendrían que cargar rápidamente.
- Funcionalidades. No se espera que todas las herramientas ofrezcan todas las funcionalidades, ni que estas sean las únicas que tendrá finalmente el Web. Entre otras: Editor de texto WYSIWYG a través del navegador, Herramienta de búsqueda, Comunicación entre los usuarios (foros, correo electrónico, chat). Noticias. Artículos. Ciclo de trabajo (*workflow*) con diferentes perfiles de usuarios y grupos de trabajo. Fechas de publicación y caducidad. Web personales. Carga y descarga de documentos y material multimedia. Avisos de actualización de páginas o mensajes en los foros, y envío automático de avisos por correo electrónico. Envío de páginas por correo electrónico. Páginas en versión imprimible. Personalización según el usuario. Disponibilidad o posibilidad de traducción al catalán y al castellano. Soporte de múltiples navegadores (Internet Explorer, Netscape, etc.). Estadísticas de uso e informes.

#### **2.4.6. Sistema Manipulador de Contenidos: Drupal.**

Drupal, entre los CMS de código abierto, se encuentra entre los más populares, su comunidad de desarrolladores y usuarios se ha ampliado en los últimos tiempos y se reportan cada vez más sitios Web que se basan en este sistema.

Con Drupal se pueden construir sitios Web dinámicos que permiten a usuarios

individuales y grupos de usuarios, publicar, administrar y organizar una gran variedad de contenidos. Drupal integra una gran variedad de populares servicios de los sistemas de administración de contenidos, entre ellas las herramientas de discusión. Como es un software *open source* el proyecto es mantenido y desarrollado por una comunidad. Su localización en Internet es [www.drupal.org](http://www.drupal.org). Paralelamente ha surgido una comunidad hispana, cuyo centro en Internet es [www.drupalhispano.org](http://www.drupalhispano.org) en ambos sitios se encuentran amplias informaciones de este CMS.

Drupal es totalmente configurable, de tal manera que el administrador de un sitio puede activar o desactivar diferentes características y establecer configuraciones que cambian el aspecto y funcionalidad del sitio. Está diseñado para ser fácilmente extensible a través de “módulos”, que no son más que bloques de código que proveen una funcionalidad extra o mejoras. Algunos módulos vienen con todas las instalaciones de Drupal (módulos del “núcleo”), mientras que otros pueden ser instalados y descargados individualmente del sitio Web de Drupal (módulos “contribuidos”).

Entre las ventajas reconocidas de Drupal está que es posible crear sitios Web de mediana complejidad de una forma mas rápida que utilizando otros CMS. En el caso de la construcción de sitios corporativos en Intranets utilizando Drupal, se reconoce como ventaja que tiene un sistema flexible de permisos en la definición de roles [FEA05].

En el sitio <http://www.cmsmatrix.org/matrix> se brinda una herramienta donde se comparan distintos CMS por parámetros. A continuación se reproducen algunos resultados, con los cuales se puede valorar las diferencias entre cada uno de ellos, basado en lo expuesto en este epígrafe.

Producto	<u>Drupal</u> 4.6.0	<u>Mambo</u> 4.5.2	<u>Moodle</u> 1.4	<u>PHP</u> <u>Nuke 6</u>	<u>Plone</u> 2.0.5	<u>Sharepoint Portal</u> <u>Server 2001</u>
Ultima Actualización	5/25/05	4/26/05	9/2/04	9/25/02	5/11/05	12/6/02
Requerimientos del sistema	Drupal	Mambo	Moodle	PHP Nuke	Plone	Sharepoint Portal Server
Servidor Aplicacion	PHP 4.3.3+	PHP 4.1.2+	PHP 4.1.2+	mod_php	Zope	IIS
Costo	Free	Free	0	Free	Free	\$30,000
	MySQL,		MySQL,	MySQL Postgres,		

<b>Bases de Datos</b>	Postgres	MySQL	Postgres	mSQL, Interbase, Sybase	Zope	MSSQL
<b>Licencia</b>	GNU GPL	GNU GPL	GNU GPL	GNU GPL	GNU GPL	Propietario
<b>Sistema Operativo</b>	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Windows
<b>Lenguaje de Programación</b>	PHP	PHP	PHP 4.1.0 or later	PHP	Python	ASP, VB, Visual C++
<b>Web Server</b>	Apache, IIS	Apache, IIS	Cualquiera	Apache, IIS	Apache, IIS, Zope	IIS
<b>Comunidad Desarrolladores</b>	Sí	Sí	Sí	Sí	Sí	No
<b>Ayuda Online</b>	Sí	Sí	Sí	No	No	Sí
<b>Editor WYSIWYG</b>	Se puede agregar	Sí	Sí	No	Sí	No
<b>Temas</b>	Sí	Sí	Sí	Sí	Sí	No
<b>Blog</b>	Sí	Sí	No	Sí	Sí	No
<b>Chat</b>	Se puede agregar	Se puede agregar	Sí	No	Se puede agregar	No
<b>Foros de Discusión</b>	Sí	Se puede agregar	Sí	Sí	Sí	Sí
<b>FAQ</b>	Sí	Sí	Sí	Sí	Se puede agregar	No
<b>Búsquedas</b>	Sí	Sí	Sí		Sí	
<b>Mapa del Sitio</b>	Se puede agregar	No			Se puede agregar	
<b>Contribuciones de los usuarios</b>	Sí	Sí	Sí	Sí	Sí	No

### 2.5. Metodología RUP. (Proceso Unificado del Rational).

RUP es el acrónimo de *Rational Unified Process*, es una metodología para el desarrollo de software propuesta por *Rational* basada en UML.

RUP organiza el desarrollo de un software, asegura la producción de software de calidad dentro de plazos y presupuestos predecibles. Dirigido por casos de uso, centrado en la arquitectura, iterativo (mini-proyectos) e incremental (versiones).

RUP, es una guía de cómo usar UML de la forma más efectiva. RUP pretende

implementar las mejores prácticas actuales en ingeniería de software: Desarrollo iterativo del software, Administración de requerimientos, Uso de arquitecturas basadas en componentes, Modelación visual del software, Verificación de la calidad del software y Control de cambios.

### **2.5.1. Lenguaje de Modelación Unificado (UML).**

El Lenguaje de Modelación Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software [JAC00]. Se basa en las anteriores especificaciones Booch, Rumbaugh y Coad-Yourdon.

Divide cada proyecto en un número de diagramas que representan las diferentes vistas del proyecto. Estos diagramas juntos son los que representan la arquitectura del proyecto [JAC00].

UML es ahora un estándar, es el resultado de la unificación de las tres opciones anteriores más populares en el mercado que fueron propuestas por Booch, Rumbaugh y Jacobson. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otra rama.

### **2.5.2. Herramienta Rational Rose.**

*Rational Rose* es la herramienta CASE desarrollada por los creadores de *UML* (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases entregables. El navegador UML de *Rational Rose* nos permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable.

Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción [JAC00].

## **2.6. Conclusiones.**

En este capítulo quedaron expuestas las ventajas de la utilización del software libre, así como sus características generales. Se analizaron además las ventajas del uso de PHP como lenguaje de programación, MySQL como Gestor de Bases de Datos y Drupal como sistema Manipulador de Contenidos. Lo que permitió utilizar estas herramientas en la creación de un sistema de soporte de software que está acorde con la tendencia internacional y las políticas nacionales en el desarrollo de productos de software bajo la concepción de software libre. Además se describieron las características de la metodología y la herramienta utilizadas en el análisis y diseño del sistema.

## **Capítulo 3. Descripción de la solución propuesta.**

---

En este capítulo se presentan las reglas del negocio propuestas por el cliente, la descripción de los procesos de negocio, que en este caso se limita a un solo proceso, el diagrama de clases del modelo de negocio, los requerimientos funcionales y no funcionales, la descripción del sistema propuesto, incluyendo definición de los actores del sistema propuestos, la identificación y descripción de los casos de uso del sistema, así como el diagrama de casos de uso del sistema.

### **3.1. Reglas del negocio.**

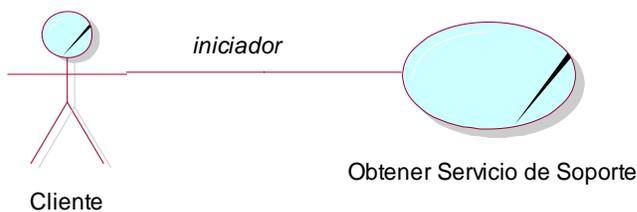
- La realización de las actividades está orientada a roles: Usuarios, Administrador, Cliente, Desarrollador, Receptor, Especialista.
- Una vez que un usuario del sistema explore uno de los productos desarrollados por la Dirección de Informatización se convierte en usuario cliente del sistema con beneficios específicos.
- El grupo de soporte está conformado por tres niveles: Receptores, Especialistas y Desarrolladores.
- Solo los Clientes del grupo de soporte emiten inquietudes al grupo.
- La primera interacción de los clientes con el grupo de soporte es mediante los Receptores.
- Solo los Receptores y Especialistas interactúan con los clientes.

### 3.2. Descripción de los procesos del negocio propuestos.

#### 3.2.1. Identificación y justificación de actores.

Actores	Justificación
Cliente	El cliente interviene en la solicitud de servicio de soporte, iniciando este proceso que concluye cuando recibe la respuesta a esta solicitud.
Trabajadores	Justificación
Desarrollador	El desarrollador es el que recibe la solicitud de soporte de los clientes, se encarga de darle solución y emitir una respuesta para el cliente.

#### 3.2.2. Diagrama de casos de uso del negocio.

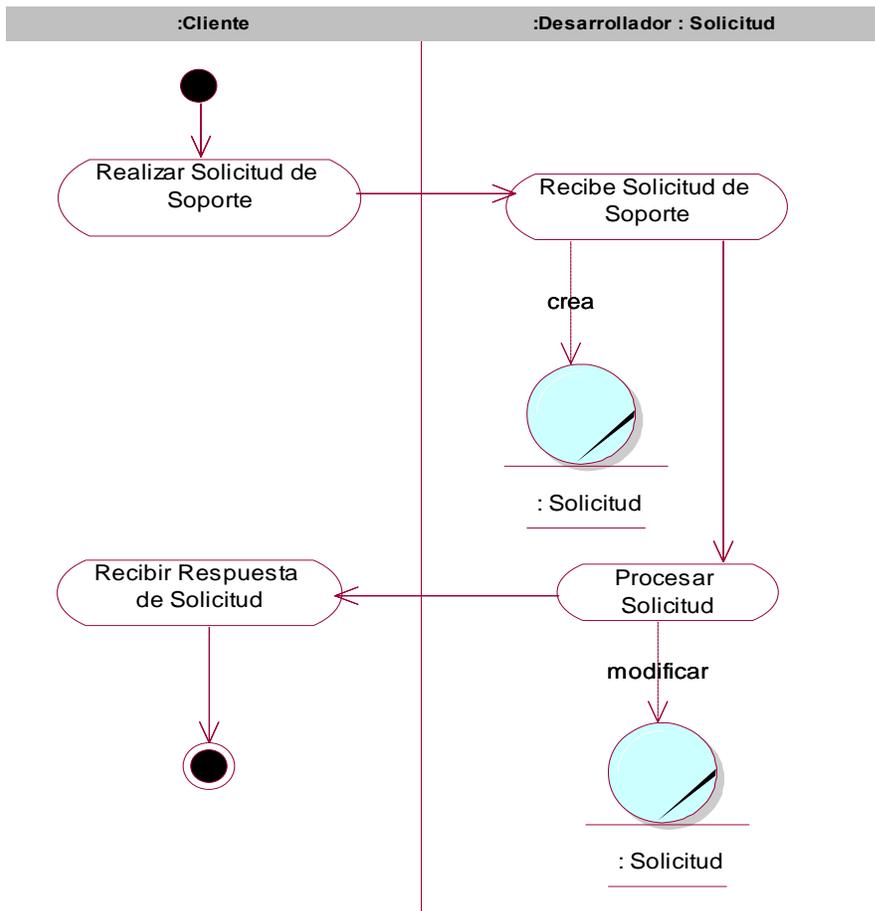


#### 3.2.3. Caso de Uso del Modelo de negocio: Solicitar Servicio de Soporte.

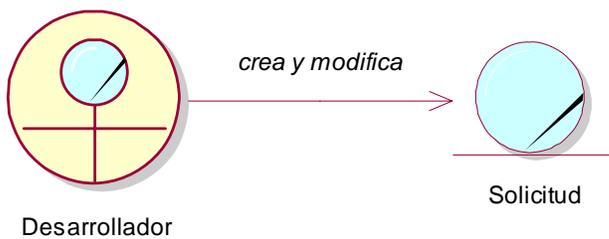
Existe un único proceso dentro del negocio: Obtener Servicio de Soporte, el que cuenta con varias etapas para su realización, como son, solicitud de un servicio de soporte, para uno de los productos que se mantiene en explotación por parte de un cliente, en la propia Dirección de Informatización o por algún medio de comunicación (teléfono, jabber, correo electrónico), recepción y creación de la solicitud por parte de un desarrollador, así como el procesamiento de la misma dando al final una solución al cliente del problema planteado. En el diagrama de actividades, todas las actividades serán objeto de automatización.

CUN-1	Solicitar servicio de Soporte
<b>Actores:</b>	Cliente
<b>Propósito:</b>	Evacuar inquietudes del cliente.
<b>Descripción:</b>	
El caso de uso se inicia cuando el cliente solicita recibir soporte para uno de sus productos, el desarrollador recibe esta solicitud, se encarga de darle respuesta y enviársela al cliente.	

### 3.2.3.1. Diagrama de Actividades: Solicitar Servicio de Soporte.



### 3.3. Diagrama de clases del modelo de objetos.



(from Business Use-Case Model)

### 3.4. Requerimientos funcionales.

R1. Mostrar usuario autenticado del sitio.

- Se muestra el nombre de usuario y la dirección electrónica

R2. Crear un nuevo usuario autenticado.

R3. Editar un usuario autenticado existente.

- R4. Eliminar un usuario autenticado existente.
- R5. Mostrar la lista de todos los productos desarrollados.
  - Se muestra el nombre del producto y la descripción.
- R6. Permitir que se agregue un nuevo producto
- R7. Editar o eliminar un producto existente.
- R8. Mostrar lista de versión(es) de un producto.
  - Se muestra el nombre de la versión, la descripción y la fecha de salida.
- R9. Agregar una nueva versión para un producto determinado.
- R10. Editar o eliminar una versión existente.
- R11. Mostrar una lista con todos los roles existentes.
  - Se muestran los nombres de los roles
- R12. Agregar un nuevo rol
- R13. Editar un rol existente.
- R14. Eliminar un rol existente.
- R15. Mostrar una lista con todos los clientes existentes.
- R16. Asignar a un cliente su área correspondiente.
- R17. Asignar a un cliente los productos que su área mantiene en explotación.
- R18. Mostrar una lista con todos los desarrolladores existentes.
- R19. Asignar a un desarrollador los productos en los que él trabajó.
- R20. Enviar solicitudes de Soporte.
  - Se envían los datos del cliente, datos del producto y el contenido de la inquietud
- R21. Enviar mensajes entre los trabajadores.
- R22. Generar y enviar respuestas intermedias y finales.
- R23. Permitir la asignación de una solicitud de soporte de un cliente a los trabajadores.
- R24. Mostrar inquietudes dado un cliente en un intervalo de tiempo determinado.
- R25. Obtener mensajería asociada a una inquietud determinada.
- R26. Obtener respuestas intermedias y respuesta final de una inquietud.
- R27. Mostrar estadísticas de los trabajadores directos con respecto a sus solicitudes asignadas en un determinado intervalo de tiempo.
  1. Total de inquietudes con respuesta final para cada trabajador.

2. Total de inquietudes asignadas para cada trabajador.
  3. Porcentaje que representan las solicitudes respondidas con respecto al total.
- R28. Mostrar una lista de trabajadores directos a los que se les haya asignado inquietudes en un intervalo de tiempo determinado.
- R29. Mostrar lista de áreas afectadas con solicitudes en un intervalo de tiempo dado.
1. Total de inquietudes realizadas por cada área.
  2. Total de inquietudes que poseen respuesta final, realizadas por cada área.
  3. Porcentaje de satisfacción para cada área: Porcentaje que representa la cantidad de inquietudes con respuesta final respecto al total de inquietudes realizadas.
- R30. Permitir a los usuarios del sitio recibir información desde los foros de discusión (todos los usuarios), además de la:
1. Participación en los foros (usuario autenticado).
  2. Creación de contenedor de foro (administrador).
  3. Creación de nuevo foro (usuario autenticado).
- R31. Permitir a los usuarios del sitio recibir información a través de las preguntas más frecuentes (FAQ).
- R32. Permitir autenticación.
- R33. Permitir que los usuarios autenticados puedan cambiar su clave de acceso.

### ***3.5. Requerimientos no funcionales.***

#### **3.5.1. Requerimientos de Apariencia o interfaz externa.**

Con la interfaz interactúan los usuarios no autenticados y los usuarios autenticados (usuarios externos que soliciten autenticación, además de los clientes y los miembros del grupo de soporte donde se incluyen los de recepción, especialistas, el jefe del Grupo de Soporte y el Administrador; así como los desarrolladores) debido a esta variedad de usuarios, la interfaz debe ser amigable y de una estructura sencilla que le permita al cualquier usuario del sitio identificar las diferentes opciones dentro del mismo.

El sistema debe permitir la personalización de la interfaz para un usuario autenticado,

en el sentido de seleccionar el fondo y la estructura básica en la presentación, a consideración de la dirección del grupo o de propio administrador.

### **3.5.2. Requerimientos de Usabilidad.**

El sitio podrá ser visitado por usuarios no registrados, o sea por cualquier usuario de la intranet. Accediendo a los servicios de lectura del desarrollo de los Foros de Discusión, acceso a las FAQ y consulta a la base de preguntas y respuestas del sitio. Para otras opciones de soporte como es el lanzar un nuevo tema de discusión al foro y realizar una pregunta o inquietud al grupo de soporte, sí se necesita ser un usuario autenticado.

Para tener acceso al sitio propuesto de soporte de software será necesario poseer acceso a la intranet de la UCI.

La herramienta está desarrollada en el idioma español por lo que es necesario que el usuario del sistema domine dicho idioma.

### **3.5.3. Requerimientos de Rendimiento.**

Debe estar desarrollada la herramienta de forma tal que garantice un rápido acceso a las páginas del sitio. El tiempo de respuesta debe ser el mínimo posible a la hora de consultar bases de información del sistema.

### **3.5.4. Requerimientos de Soporte.**

Para garantizar el soporte a los clientes de esta herramienta, se utilizará la propia herramienta, ya que al ser instalada en el grupo de soporte de software de la Dirección de Informatización, se establecerá un contenedor de foro con el tema “Sistema de Soporte de Software”, en el cual se propondrán los foros relacionados con las inquietudes del sistema como tal.

### **3.5.5. Requerimientos de Portabilidad.**

La herramienta propuesta podrá ser usada bajo cualquier sistema operativo, para su implementación se usaron Herramientas de Programación y Gestión de Bases de Datos que son multiplataforma.

### **3.5.6. Seguridad.**

Se debe dar la posibilidad de aplicarle las políticas de seguridad existentes en el centro.

Confiabilidad: la información manejada por el sistema está protegida de acceso no autorizado.

Integridad: la información manejada por el sistema será objeto de protección contra estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos.

Disponibilidad: a los usuarios autorizados se les garantizará el acceso a la información y los dispositivos o mecanismos utilizados. No deberá ocultarse o retrasarse a los usuarios la obtención de datos deseados en el momento deseado.

En el sistema, deben estar bien definidos los roles de los diferentes usuarios, especificando para cada uno de ellos los permisos que posee en la manipulación del software. Los usuarios autenticados (específicamente, Jefe del Grupo de Soporte, Administrador, Receptores, Especialistas, Desarrolladores y Clientes) poseen opciones diferentes vinculadas a diferentes acciones que pueden realizar. Mientras que los usuarios no autenticados pueden visitar y navegar por el sitio, pero con otras opciones, así como el resto de los usuarios autenticados.

### **3.5.7. Legales.**

Debe cumplir con las normas, leyes y políticas de seguridad establecidas por el país y por la institución.

### **3.5.8. Confiabilidad.**

Debe tener servidores de respaldo de la Base de Datos para casos de problemas en el servidor principal, como puede ser la pérdida de la información en caso de fallos en el momento de hacer operaciones sobre los datos (insertar, eliminar o modificar), permitiendo su recuperación.

### **3.5.9. Ayuda y documentación en línea.**

La herramienta debe contar con un contenedor de foro donde se publicarán los foros de temas relacionados con el uso de la aplicación, allí se esclarecerán dudas sobre su uso. Además brindará una opción para que el usuario encuentre una explicación de cómo navegar por él, así como las facilidades que le brinda.

### **3.5.10. Requerimientos de Software.**

En los equipos que hagan función de servidor, independientemente del sistema operativo implantado, se necesita: servidores Web (Apache o IIS que soporte el lenguaje PHP), de Base de Datos (MySQL). En las computadoras de los usuarios y del grupo de soporte sólo se requiere de navegador para acceder a la Intranet de la UCI.

### **3.5.11. Requerimientos de Hardware.**

Se requiere de un servidor de 128 MB de RAM como mínimo y 6 GB de capacidad del disco duro, todas las computadoras implicadas, tanto para la administración como las de los usuarios, deben estar conectadas a una red y tener al menos 64 MB de RAM.

### **3.5.12. Restricciones en el diseño y la implementación.**

Para ser consecuente con el planteamiento de hacer una herramienta para brindar Soporte de Software, que se base en el software libre, es necesario utilizar para su implementación lenguajes de programación que sean multiplataforma, en este caso el PHP, así mismo, es necesario usar un Sistema Gestor de Bases de Datos multiplataforma, como es el caso de MySQL.

Para garantizar una documentación actualizada del sistema se utiliza para realizar el análisis y el diseño del sistema la metodología RUP con el lenguaje UML (Unified Modelling Language). Como herramienta de apoyo a este Lenguaje de Modelación se

utiliza Rational Rose.

### **3.6. Descripción del sistema propuesto.**

#### **3.6.1. Concepción general del sistema.**

Se desarrolla una herramienta para brindar Soporte de Software en la UCI, con el objetivo de lograr mayor satisfacción a las necesidades de los clientes. Para determinar las opciones con las que contaría la herramienta, se visitaron sitios de soporte de software en Internet y se estudiaron otros softwares que tienen este mismo objetivo.

El sistema propuesto permite:

- **Actualizar Información:** Mantener actualizada la información referida a los productos y sus versiones; la referente a los clientes.
- **Responder Solicitud de los clientes:** Al añadir una inquietud al sistema, un Recepcionista se encarga de ofrecer una respuesta, en caso de que no le sea posible, se le asigna a un Especialista, el cual se encargará de resolverla contando con el apoyo, si es necesario, de un desarrollador.
- **Conocer estadísticas:** Se calculan estadísticas de desempeño de los trabajadores del grupo de soporte y de los clientes. Estadísticas de las incidencias por grupos de resolución, tiempos de resolución, usuarios y otros parámetros. Las mismas tienen como objetivo poder realizar un análisis de la actividad, conllevando al mejoramiento del servicio y la operabilidad de los usuarios.
- **Reconocer usuarios autenticados:** El sistema reconoce los usuarios autenticados y los roles que desempeñan.
- **Disponer de una base de preguntas y respuestas** con información para la toma de decisiones del equipo de soporte y apoyo a los equipos de desarrollo.

#### **3.6.2 Definición de los actores del sistema a automatizar:**

<b>Actores</b>	<b>Breve Descripción</b>	<b>Requerimientos Asociados</b>
----------------	--------------------------	---------------------------------

Administrador	Representa a una persona que administra los datos de los usuarios y productos. Un administrador utiliza el sistema para agregar, modificar o eliminar estos datos, así como listar la información referente a estos.	R6, R7, R9-R19
Receptor	Representa a una persona que recepciona, da solución o asigna a otro nivel las inquietudes formuladas por los clientes. Utiliza el sistema para responder una inquietud, apoyándose en la base de conocimientos o para asignar la inquietud a un trabajador de Nivel 2.	R23
Especialista	Representa a una persona que da solución a las inquietudes formuladas por los clientes. Utiliza el sistema para responder una inquietud apoyándose en la base de conocimientos y/o en los trabajadores de Nivel 3.	
Desarrollador	Representa a una persona que responde a consultas realizadas por un trabajador de Nivel 2. Utiliza el sistema para intercambiar información especializada con un trabajador de Nivel 2 que solicite ayuda para dar solución a la inquietud de un cliente.	
Cliente	Representa a una persona que solicita una respuesta a un problema determinado. Utiliza el sistema para adquirir información mediante el envío de una inquietud al grupo de soporte.	R20, R24, R26
Jefe de Grupo de Soporte	Representa a una persona que controla el funcionamiento del grupo de soporte. Utiliza el software para obtener reportes, así como datos estadísticos relacionados con los trabajadores y clientes del grupo.	R24-R29
Usuarios autenticados	Representa a una persona que obtiene información general del grupo y sus actividades. Utiliza el sistema para crear o utilizar foros de discusión dentro de los contenedores definidos.	R32, R33
Usuarios no autenticados	Representa a una persona que obtiene información del grupo y sus actividades. Utiliza el software para consultar foros de discusión y	

	consultar las FAQ.	
Usuario	Representa un actor abstracto dentro del sistema del que heredan los Usuarios autenticados y los no autenticados.	R1-R5, R8, R30, R31
Trabajador directo	Representa un actor abstracto dentro del sistema del que heredan los Receptores y los Especialistas.	R22
Trabajador	Representa un actor abstracto dentro del sistema del que heredan los Desarrolladores.	R21



<b>CU-1</b>	<b><i>Gestionar usuario</i></b>
<b>Actores:</b>	Usuario
<b>Propósito:</b>	Actualizar cuenta de usuario.
<b>Descripción:</b> El caso de uso se inicia cuando el Usuario solicita crear, editar (incluye cambio de contraseña) o eliminar un usuario autenticado (en el caso del Administrador puede editar o eliminar cualquiera de los existentes). De acuerdo a su requerimiento selecciona la acción a realizar, siguiendo los pasos correspondientes, hasta crear, actualizar o eliminar el registro de usuario.	
<b>Referencias:</b>	R1, R2, R3, R4, R33
<b>Precondiciones:</b>	Existencia de usuario registrado en caso de corresponder a requerimientos de edición o eliminación.
<b>Poscondiciones:</b>	Si el requisito asociado es crear un nuevo usuario: debe quedar registrado el usuario.  Si el requisito asociado es editar usuario: debe quedar actualizado el registro.  Si el requisito asociado es eliminar usuario: debe quedar eliminado el usuario seleccionado.

<b>CU-2</b>	<b><i>Listar productos</i></b>
<b>Actores:</b>	Usuario.
<b>Propósito:</b>	Que los usuarios tengan la posibilidad de ver todos los productos desarrollados en la Dirección de Informatización.
<b>Descripción:</b> El caso de uso se inicia cuando un usuario solicita listar los productos desarrollados, mostrándosele el nombre y la descripción del producto y una lista de versiones del mismo con su nombre, descripción y fecha de salida.	
<b>Referencias:</b>	R5, R8
<b>Precondiciones:</b>	Existencia de productos en la Base de Datos.

<b>CU-3</b>	<b><i>Gestionar productos</i></b>
<b>Actores:</b>	Administrador.
<b>Propósito:</b>	Que el administrador del sistema pueda realizar operaciones de gestión sobre los productos y sus versiones (adicionar, modificar o eliminar)
<b>Descripción:</b> El caso de uso se inicia cuando un administrador solicita acceder a las opciones de crear, modificar o eliminar un producto o una versión de un producto De acuerdo a su requerimiento selecciona la acción a realizar, siguiendo los pasos correspondientes, hasta crear, actualizar o eliminar el registro del producto o la versión.	
<b>Referencias:</b>	R6, R7, R9, R10
<b>Precondiciones:</b>	Administrador autenticado satisfactoriamente por el sistema.  Existencia de productos en la Base de Datos para la edición o eliminación.
<b>Poscondiciones:</b>	Producto o versión registrada en la Base de Datos, en caso de que el administrador seleccione la opción de agregar producto o versión.  Registro del producto o versión actualizado en la Base de Datos, en caso de que el administrador seleccione la opción de modificar producto o versión.  Si el requisito asociado es eliminar producto o versión, debe quedar eliminado el producto o versión de la Base de Datos.

<b>CU-4</b>	<b><i>Gestionar roles</i></b>
<b>Actores:</b>	Administrador.
<b>Propósito:</b>	Mostrar al administrador la lista de roles para seleccionar operaciones de gestión.
<b>Descripción:</b> El caso de uso se inicia cuando el administrador solicita mostrar la lista de roles registrados donde se accede además a las opciones de agregar, editar y eliminar un rol. De acuerdo a su requerimiento selecciona la acción a realizar, siguiendo los pasos	

<p>correspondientes, hasta crear, actualizar o eliminar el registro del rol en la Base de Datos y mostrándose finalmente la lista de roles con las operaciones realizadas.</p> <p>Una vez mostrada la lista de roles, el administrador puede verificarlos.</p> <p>Esta funcionalidad está implícita dentro de la herramienta que se utilizará en el desarrollo de la aplicación.</p>	
<b>Referencias:</b>	R11, R12, R13, R14
<b>Precondiciones:</b>	<p>Administrador autenticado satisfactoriamente por el sistema.</p> <p>Existencia de roles en la Base de Datos para el caso de edición y eliminación.</p>
<b>Poscondiciones:</b>	<p>Si el requerimiento asociado es crear un nuevo rol: debe quedar registrado el rol en la Base de Datos.</p> <p>Si el requerimiento asociado es editar rol: debe quedar actualizado el registro en la Base de Datos.</p> <p>Si el requerimiento asociado es eliminar rol: debe eliminarse el rol seleccionado de la Base de Datos.</p>

<b>CU-5</b>	<b><i>Gestionar clientes</i></b>
<b>Actores:</b>	Administrador.
<b>Propósito:</b>	Mostrar al administrador la lista de clientes para seleccionar operaciones de gestión.
<b>Descripción:</b>	<p>El caso de uso se inicia cuando el administrador solicita listar los datos de los clientes para consultarlos o para acceder a las opciones de asignarle área y productos a uno de ellos, en este caso el administrador solicita asignarle área y producto(s) a un cliente, entra los datos y se actualiza la Base de Datos. Finalmente se muestra al administrador la lista de usuarios sin modificación visible alguna.</p>
<b>Referencias:</b>	R15, R16, R17
<b>Precondiciones:</b>	<p>Administrador autenticado satisfactoriamente por el sistema.</p> <p>Existencia, en la Base de Datos, de al menos un cliente.</p>
<b>Poscondiciones:</b>	Si el administrador solicita asignar datos al cliente, debe quedar el registro del cliente actualizado en la Base de Datos.

<b>CU-6</b>	<b><i>Gestionar desarrolladores</i></b>
<b>Actores:</b>	Administrador.
<b>Propósito:</b>	Mostrar al administrador la lista de desarrolladores para seleccionar operaciones de gestión.
<b>Descripción:</b> El caso de uso se inicia cuando el administrador solicita listar los datos de los desarrolladores para consultarlos o para asignarle versiones a uno de ellos, en este caso el administrador solicita asignarle versión(es) a un desarrollador, entra los datos y se actualiza la Base de Datos.	
<b>Referencias:</b>	R18, R19
<b>Precondiciones:</b>	Administrador autenticado satisfactoriamente por el sistema. Existencia, en la Base de Datos, de al menos un desarrollador.
<b>Poscondiciones:</b>	Si el administrador solicita asignar datos al desarrollador, debe quedar el registro de desarrollador actualizado en la Base de Datos.

<b>CU-7</b>	<b><i>Enviar inquietud</i></b>
<b>Actores:</b>	Cliente.
<b>Propósito:</b>	Permitir a un cliente realizar una pregunta respecto a un producto determinado.
<b>Descripción:</b> El caso de uso se inicia cuando un cliente, formula la inquietud acerca de un producto específico. Esta inquietud se almacena como inquietud no asignada a un especialista y como inquietud no asignada a un receptor y sin respuesta final en la Base de Datos.	
<b>Referencias:</b>	R20
<b>Precondiciones:</b>	Cliente autenticado satisfactoriamente por el sistema. Existencia del producto del que se formula la inquietud.
<b>Poscondiciones:</b>	Debe quedar insertado el registro inquietud en la Base de Datos.

<b>CU-8</b>	<b><i>Responder inquietud</i></b>
<b>Actores:</b>	Trabajador directo
<b>Propósito:</b>	Permitir a un receptor o especialista responder una inquietud, asociada a un cliente, de un producto determinado.
<b>Descripción:</b>	
<p>El caso de uso se inicia cuando un receptor pretende responder la inquietud del cliente (automáticamente se asigna la inquietud al receptor), si es capaz de responder queda almacenada la respuesta, indicándose que es respuesta final.</p> <p>Si el receptor no puede responder totalmente la inquietud se la asigna a un especialista (puede asignarse a más de un especialista) y genera una respuesta intermedia por si el cliente solicita el estado de esta consulta. Esta respuesta intermedia se almacena, indicándose que no es respuesta final.</p>	
<b>Referencias:</b>	R22, R23
<b>Precondiciones:</b>	<p>Receptor o especialista autenticado satisfactoriamente por el sistema.</p> <p>Existencia de la inquietud sin respuesta final.</p>
<b>Poscondiciones:</b>	<p>Deben quedar almacenadas todas las respuestas intermedias en caso de existir y la respuesta final.</p> <p>Debe quedar asignada por el sistema al receptor la inquietud una vez seleccionada y en caso de este no pueda solucionarla, debera asignarla a un especialista</p>

<b>CU-9</b>	<b><i>Enviar mensaje por inquietud</i></b>
<b>Actores:</b>	Trabajador
<b>Propósito:</b>	Evacuar dudas o dificultades a la hora de responder una solicitud de soporte.
<b>Descripción:</b>	
<p>El caso de uso es iniciado por un especialista, un receptor o un desarrollador. Si el especialista no puede dar una solución final a la inquietud intercambia mensajería con un desarrollador de ese producto hasta que sea capaz de enviar una respuesta intermedia o</p>	

final al cliente. En la Base de Datos quedaran almacenados todos estos mensajes. De igual forma pueden intervenir los receptores en estas acciones, solo en casos requeridos.	
<b>Referencias:</b>	R21
<b>Precondiciones:</b>	Trabajador autenticado satisfactoriamente por el sistema.  Existencia de inquietudes realizadas.

<b>CU-10</b>	<b><i>Evaluar inquietud</i></b>
<b>Actores:</b>	Jefe Grupo de Soporte
<b>Propósito:</b>	Que el Jefe de Grupo de Soporte valore el trabajo realizado por el grupo acerca de una inquietud formulada, teniendo en cuenta todas las repuestas emitidas y la mensajería intercambiada por los trabajadores.
<b>Descripción:</b> El caso de uso es iniciado por el Jefe de Grupo de Soporte o un receptor cuando solicitan que se muestren todas las inquietudes formuladas por un cliente en un intervalo de tiempo determinado, donde escoge dar seguimiento a una inquietud específica. Se mostrarán de la inquietud, todas las respuestas intermedias y la respuesta final si están dadas y la mensajería Trabajadores en caso de que existan.	
<b>Referencias:</b>	R24, R25, R26
<b>Precondiciones:</b>	Jefe de Grupo de soporte autenticado satisfactoriamente por el sistema.  Existencia de inquietudes realizadas por el cliente en el intervalo de tiempo especificado.

<b>CU-11</b>	<b><i>Obtener respuestas de inquietud</i></b>
<b>Actores:</b>	Cliente
<b>Propósito:</b>	Que el cliente conozca el estado de una inquietud realizada mostrándose todas las respuestas intermedias y en caso de existir, la respuesta final de la inquietud.
<b>Descripción:</b> El caso de uso es iniciado por el cliente cuando solicita que se muestren todas las	

<p>inquietudes formuladas por él en un intervalo de tiempo determinado, donde escoge obtener respuestas de una inquietud específica. Se mostrarán de la inquietud, todas las respuestas intermedias y respuesta final si están dadas.</p>	
<b>Referencias:</b>	R26
<b>Precondiciones:</b>	<p>Cliente autenticado satisfactoriamente por el sistema.</p> <p>Existencia de inquietudes realizadas por el cliente en cuestión en el intervalo de tiempo especificado.</p>

<b>CU-12</b>	<b><i>Obtener estadísticas de los trabajadores</i></b>
<b>Actores:</b>	Jefe Grupo de Soporte
<b>Propósito:</b>	Que el Jefe de Grupo de Soporte valore la capacidad de trabajo y eficiencia de cada trabajador del grupo.
<b>Descripción:</b>	
<p>El caso de uso es iniciado por el Jefe del Grupo de Soporte cuando solicita obtener la lista de trabajadores con cantidad de preguntas respondidas (con respuesta final) en un intervalo de tiempo dado agrupados por el rol que desempeña (receptor o especialista) además de obtener, en caso de tratarse de un especialista, el total de las asignadas sin respuesta final además del por ciento de eficacia del especialista (por ciento que representa el total de preguntas respondidas con respecto a las asignadas). También deben mostrarse en todos los casos el contenido de las inquietudes (incluida la respuesta final en caso que exista).</p>	
<b>Referencias:</b>	R27, R28
<b>Precondiciones:</b>	Jefe de Grupo de Soporte autenticado satisfactoriamente por el sistema.

<b>CU-13</b>	<b><i>Obtener estadísticas de las áreas</i></b>
<b>Actores:</b>	Jefe Grupo de Soporte
<b>Propósito:</b>	Que el Jefe de Grupo de Soporte valore la capacidad de trabajo y eficiencia de cada trabajador del grupo.
<b>Descripción:</b>	

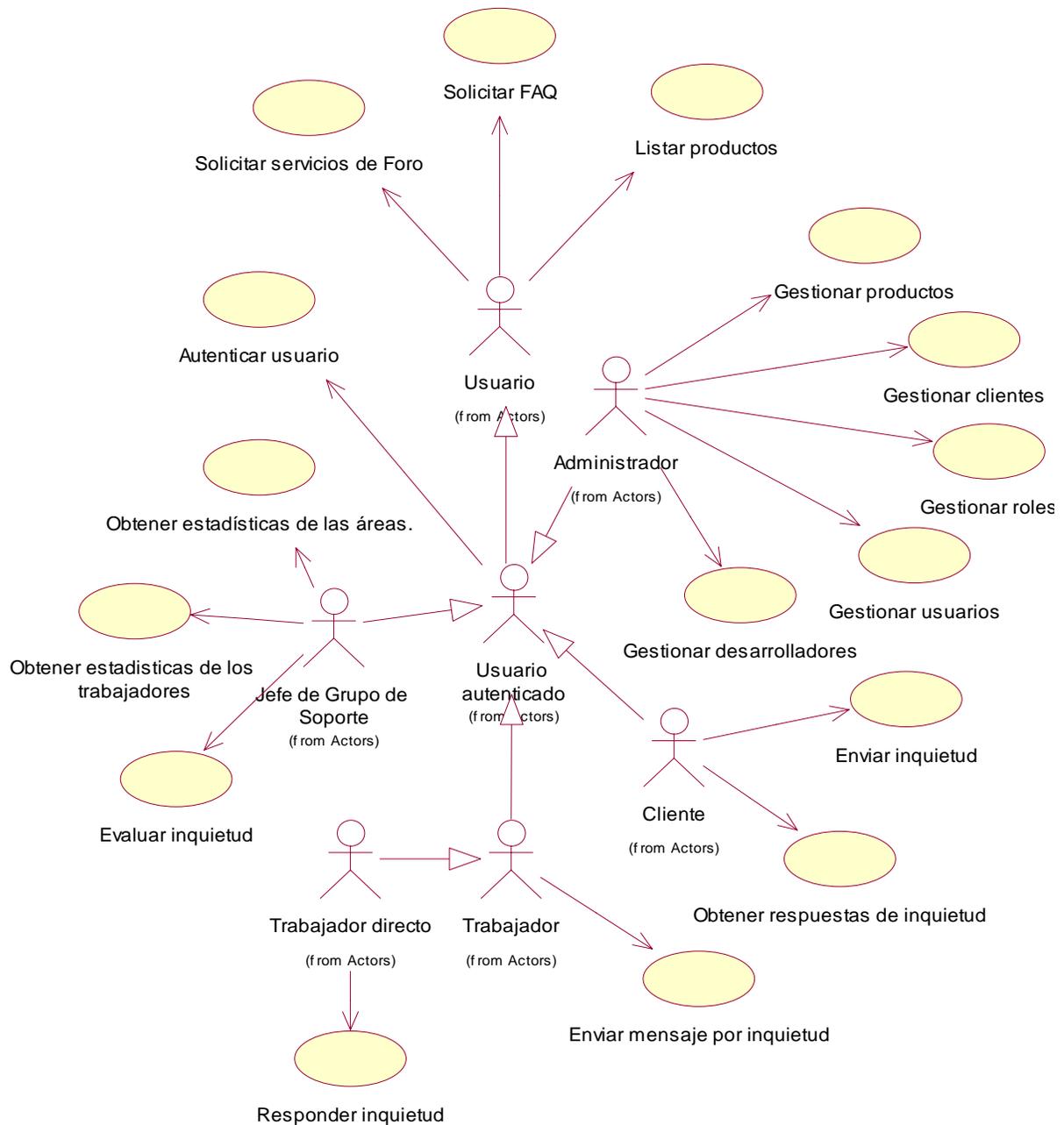
<p>El Jefe del Grupo de Soporte solicita obtener la lista de áreas con total de inquietudes formuladas, total de inquietudes formuladas con respuesta final y por ciento de satisfacción al cliente (por ciento que representa el total de inquietudes con respuesta final, respecto al total de inquietudes formuladas) en un intervalo de tiempo dado. Deberá obtenerse la relación de inquietudes y respuestas de las inquietudes con respuesta final.</p>	
<b>Referencias:</b>	R29
<b>Precondiciones:</b>	Jefe de Grupo de Soporte autenticado satisfactoriamente por el sistema.

<b>CU-14</b>	<b><i>Solicitar servicio de foro</i></b>
<b>Actores:</b>	Usuario
<b>Propósito:</b>	Que el usuario cliente pueda evacuar la mayor cantidad de inquietudes antes de formular una inquietud al grupo de soporte.
<b>Descripción:</b>	
<p>El caso de uso se inicia cuando un usuario solicita que se le muestren los contenidos de los foros existentes (todas las intervenciones) o cuando los usuarios autenticados solicitan agregar comentarios, dentro de los foros existentes, además de crear una nueva discusión o foro. O cuando el administrador solicita la operación de agregar o eliminar contenedor de foro. De acuerdo a la acción a realizar, siguiendo los pasos correspondientes, se actualiza el servicio de foro de discusión.</p>	
<b>Referencias:</b>	R30
<b>Precondiciones:</b>	<p>Administrador autenticado satisfactoriamente por el sistema en caso de tratarse de la eliminación o inserción de un contenedor de foros.</p> <p>Usuario autenticado, autenticado satisfactoriamente por el sistema en caso de tratarse de agregar comentario o proponer otro foro.</p>
<b>Poscondiciones</b>	<p>Si el usuario autenticado solicita crear nuevo foro de discusión o agregar un comentario a uno existente debe quedar registrado en la Base de Datos</p> <p>Si el requisito asociado es crear un nuevo contenedor de foro: debe quedar registrado el contenedor en la Base de Datos.</p> <p>Si el requisito asociado es eliminar contenedor: debe eliminarse</p>

	el contenedor seleccionado de la Base de Datos.
<b>CU-15</b>	<b><i>Solicitar FAQ</i></b>
<b>Actores:</b>	Usuario
<b>Propósito:</b>	Que el usuario cliente pueda evacuar la mayor cantidad de inquietudes antes de formular una inquietud al grupo de soporte.
<b>Descripción:</b> El caso de uso se inicia cuando un usuario del sistema solicita que se muestren la preguntas más frecuentes realizadas por los clientes, se muestra una lista con estas preguntas y puede accederse a la respuesta correspondiente.	
<b>Referencias:</b>	R31
<b>Precondiciones:</b>	Reiteración de solicitudes de soporte por parte de los clientes.

<b>CU-16</b>	<b><i>Autenticar usuario</i></b>
<b>Actores:</b>	Usuario autenticado
<b>Propósito:</b>	Que el usuario disponga de los beneficios del sitio según su rol, además de la seguridad de la aplicación.
<b>Descripción:</b> El caso de uso es iniciado por un usuario autenticado cuando solicita entrar al sistema, introduce su nombre de usuario, su contraseña y en caso de ser correctos el sistema lo autentica satisfactoriamente.	
<b>Referencias:</b>	R32
<b>Precondiciones:</b>	Para el caso de cambio de contraseña, debe estar registrado el usuario anteriormente.

### 3.6.4. Diagrama de Casos de Uso del Sistema.



### 3.7. Conclusiones.

Se concluye la etapa de análisis del sistema, que abarcó el estudio del negocio y la descripción de los requerimientos funcionales y no funcionales del sistema propuesto. Se realizaron los casos de uso respondiendo a las guías de la metodología utilizada y finalmente se muestra el diagrama de casos de uso del sistema donde se pueden visualizar de manera general las funcionalidades que debe poseer el sistema.

## **Capítulo 4. Construcción de la solución propuesta.**

---

En el desarrollo de este capítulo se abordarán elementos vinculados a las generalidades de los CMS, especificando algunos respecto a la arquitectura y funcionalidades que presenta Drupal. Además se representan los diagramas de clases por paquetes conformados, así como el diagrama de clases persistentes y el de modelo de datos. También se tratan o muestran los principios de diseño, estándares de codificación, diagramas de despliegue y de componentes.

### **4.1. Diagramas de clases.**

De acuerdo a que se propone utilizar el manejador de contenidos Drupal, es necesario abordar algunas características, en cuanto a comportamiento y componentes, de los CMS en general y del Drupal específicamente, para un buen entendimiento de esta parte del trabajo.

Un sistema de manejo o de administración de contenidos permite crear o modificar la información que contiene un sitio de una forma dinámica, basándose en la construcción o adjunción de módulos para la parte funcional de la aplicación y en el manejo de plantillas para la parte de diseño. Se puede agregar cuantas funcionalidades requiera su sitio de una manera sencilla, basta con descargar los módulos de Internet que respondan a la operabilidad requerida o con construirlo con una estructuración específica [SIS03].

En este capítulo se denominará Paquetes a los módulos creados e incluidos dentro de la plataforma Drupal, la que podemos considerar también como un paquete general que contendrá, además, a determinados elementos, propios del Drupal, la carpeta *modules*, la carpeta *includes*, la carpeta *themes* y la única página servidor que presenta. Esta última se encarga de hacer los llamados necesarios para la construcción de la página final, de acuerdo al diseño seleccionado y a lo que se desea mostrar de una forma personalizada (envío de argumentos en la petición), esto se logra mediante módulos que regulan el acceso y permiso a funcionalidades definidas por el administrador para cada rol dentro del sistema.

Estas agrupaciones pertenecen al núcleo lógico de la herramienta y responden tanto a los niveles de acceso a datos como a los de negocio e interfaz. Se decidió representar como paquetes a los módulos implementados por la independencia que existe entre ellos a la hora de su creación.

Al no ser objetivo en este trabajo la modelación de la herramienta que será explotada, en particular las funcionalidades que brinda y se pretenden utilizar, como son los foros de discusión, la publicación de artículos, el manejo de los usuarios del sitio, incluyendo la seguridad y los beneficios que obtendrá de acuerdo a su rol, así como su estructuración general, sino solo lo que responda a las especificidades incluidas dentro de ella, se contemplarán únicamente, en esta etapa del diseño, los módulos que será necesario desarrollar, excepto las mencionadas agrupaciones con el objetivo de ofrecer una visión lo más cercana posible de la operabilidad de la futura aplicación.

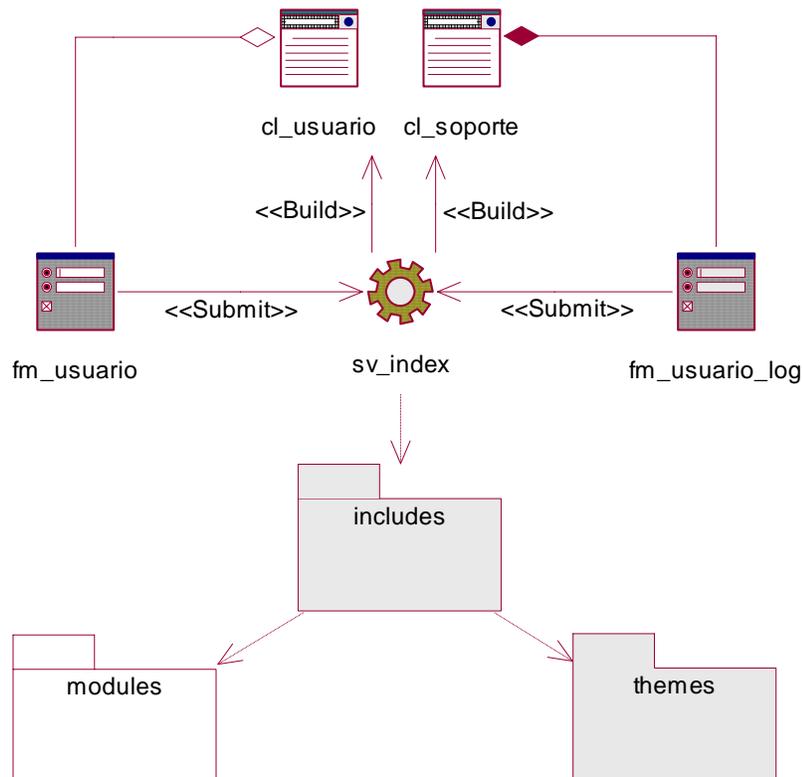
Se han de desarrollar cinco módulos: clientes, desarrolladores, productos, consultas y resultados y se interpretará a dichos módulos como paquetes, de la siguiente manera: al módulo clientes como paquete clientes, al módulo desarrolladores como paquete desarrolladores y de igual forma a los módulos productos, consultas y resultados.

Se hará uso, en esta etapa de la modelación, de las mencionadas agrupaciones propias del Drupal. En este caso se representará como el paquete *includes* a la carpeta *includes*, como el paquete *themes* a la carpeta *themes* y como el paquete *modules* a la carpeta *modules*. La carpeta *includes* del Drupal contiene un grupo de ficheros de configuración y es donde se vinculan todas las partes, la carpeta *themes* encierra al sistema de plantillas y estilos y en la carpeta *modules* aparecen una serie de módulos que conforman al núcleo funcional de la plataforma, los que no serán objeto de representación en este capítulo, además de los módulos construidos.

En lo adelante los elementos sombreados son propios del CMS Drupal.

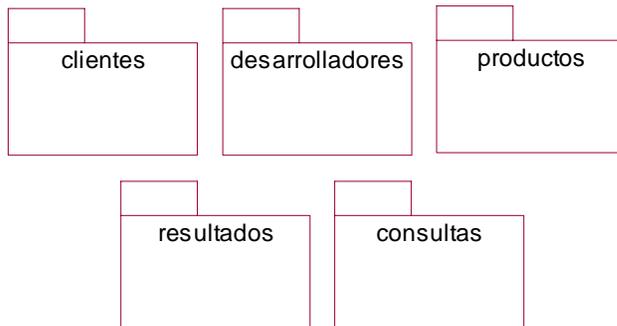
### 4.1.1. Paquete Drupal.

El CMS Drupal cuenta con una única página servidor. En todo llamado que se realiza y de acuerdo a los argumentos enviados, es que construye la página deseada, partiendo de las invocaciones que realiza a la carpeta *includes*, la que a su vez se encarga de contemplar las sesiones, el reconocimiento de la Base de Datos, la conexión a la Base de Datos, prescindiendo el programador de realizar una conexión a la Base de Datos una vez escrita una sentencia SQL en cualquier parte de la plataforma siempre que utilice estas funciones, además de la configuración de la página cliente de acuerdo al estilo y plantilla seleccionados, el idioma escogido y también de invocar a todos los módulos para la ejecución de su funcionalidades si son requeridas.



#### 4.1.1.1. Paquete modules.

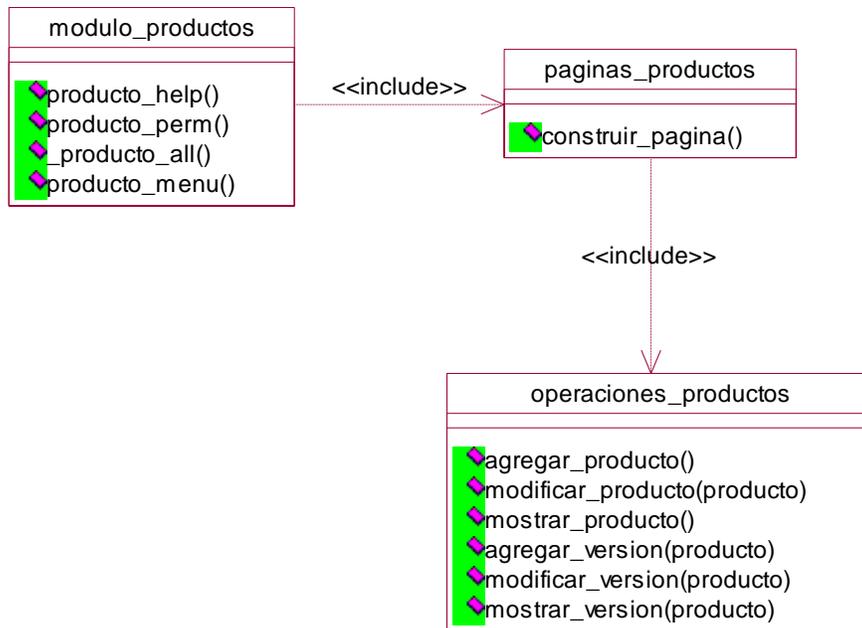
El paquete modules contiene los módulos desarrollados con funcionalidades específicas y no presentan relación alguna entre ellos.



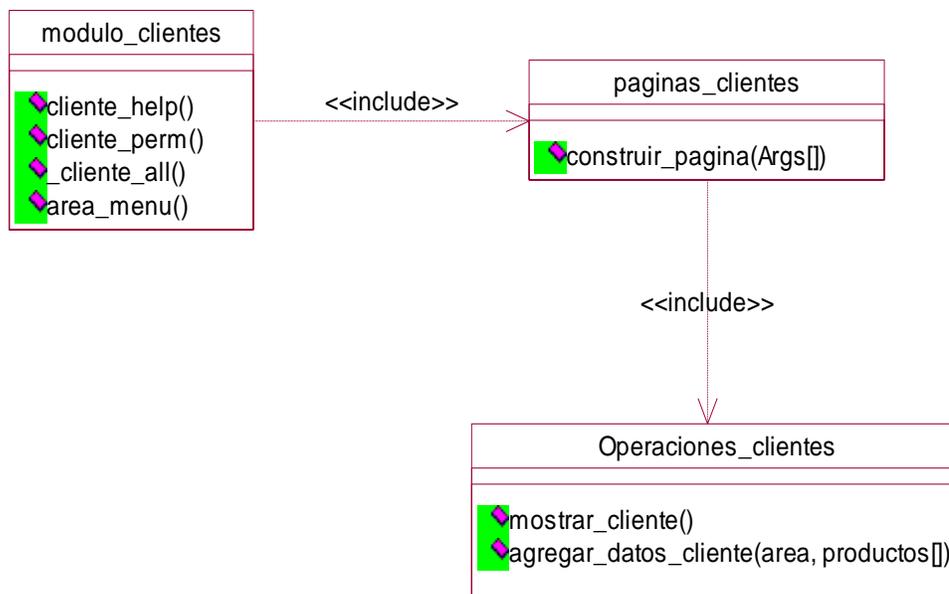
A continuación se mostrarán los diagramas de clases correspondientes a los módulos desarrollados. Existe gran similitud entre estos diagramas, gracias a la estructura que ofrece Drupal para la construcción de sus módulos.

En esencia consisten en una clase *módulo\_X* (se interpretará X como el nombre del paquete), que representa al fichero con extensión *.module* que es creado en Drupal para el reconocimiento de la estructura como módulo de la plataforma, el que contiene una arquitectura predeterminada. Además una clase *páginas\_X*, que representa a un fichero que se encarga de generar el código HTML a mostrar, a partir de determinados argumentos con que se efectuó la petición, para la construcción de la página. Por último una clase denominada *operaciones\_X*, que representa a una clase controladora.

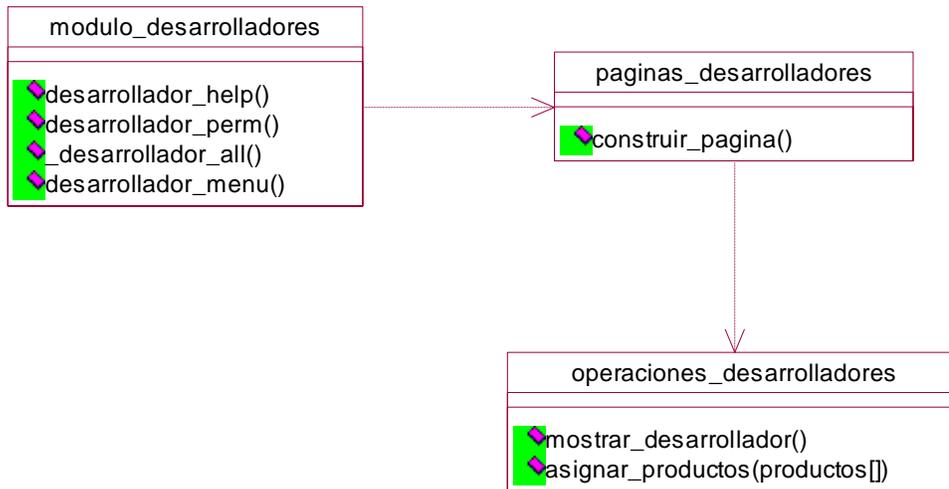
#### 4.1.1.1.1. Sub paquete productos.



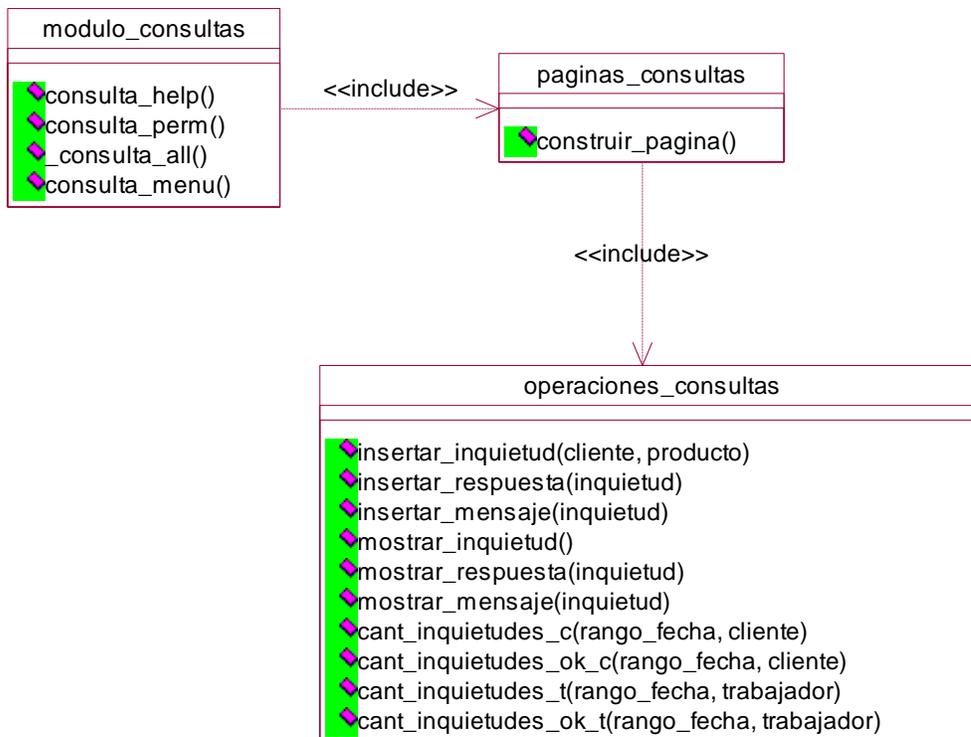
#### 4.1.1.1.2. Sub paquete clientes.



#### 4.1.1.1.3. Sub paquete desarrolladores.

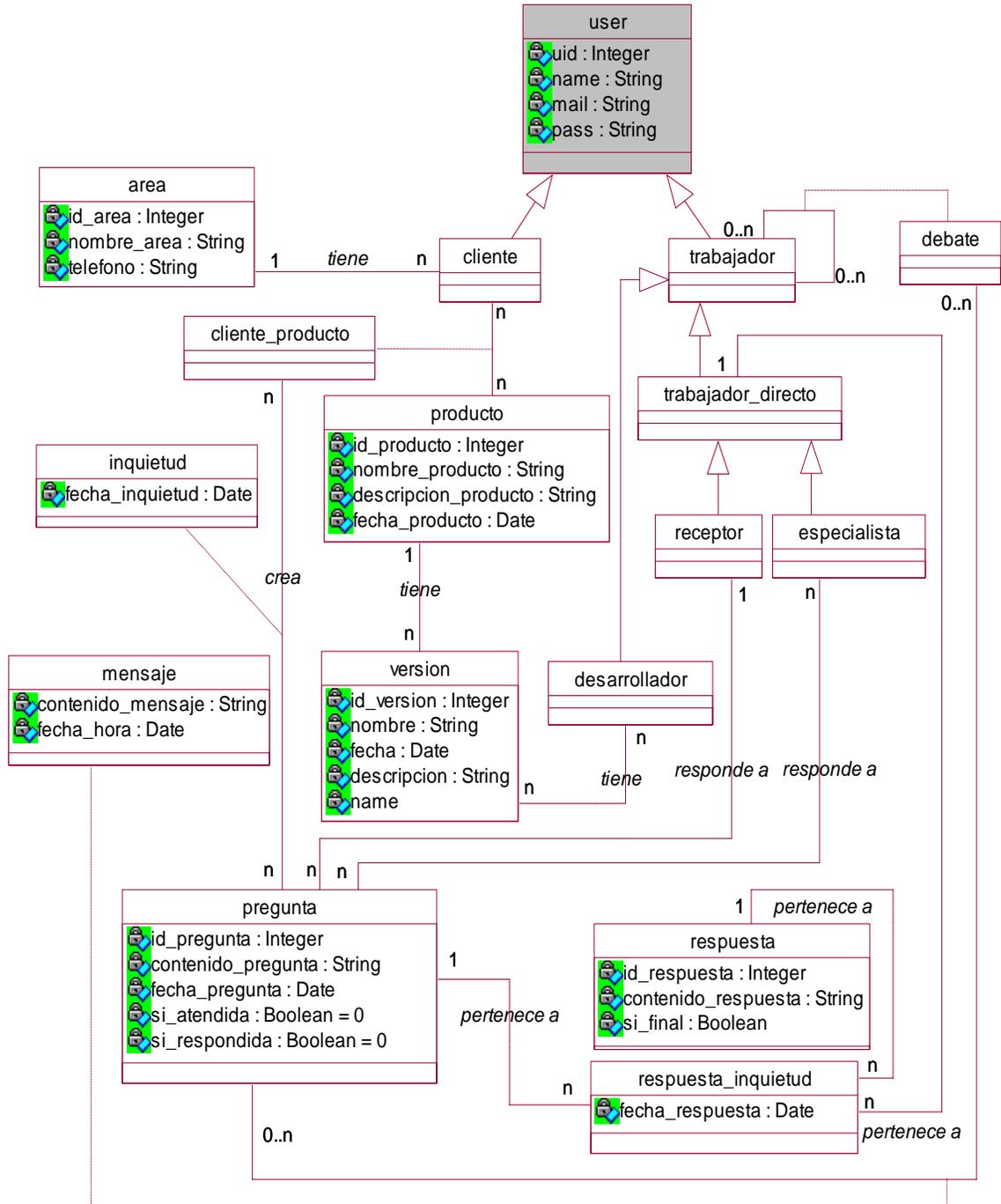


#### 4.1.1.1.4. Sub paquete consultas.



## 4.2. Diseño de la Base de Datos.

### 4.2.1. Diagrama de clases Persistentes.





estilos que el CMS Drupal ofrece. Para las opciones de administración la aplicación está diseñada para personas capacitadas, aún así no debe ser muy complejo de manipular, proporcionando una plataforma sencilla y flexible. Aprovechando las bondades de Drupal como Sistema de Gestión de Contenido se logran prácticamente todos estos requerimientos.

#### **4.3.1. Estándares en la interfaz de la aplicación.**

El sitio está diseñado para una resolución estándar de la pantalla de 800x600 píxeles, aunque el Drupal está validado para el caso en que varíe. El color de fondo del sitio es blanco, y para las imágenes y demás componentes que lo requieren se utilizó el color azul y el gris. El idioma utilizado es el español.

En todas las páginas se mantiene un menú en la parte izquierda, que permite el acceso a todas las opciones que se brindan en el sitio, además de mostrar el usuario autenticado a cada momento y en caso que no esté navegando en el sitio un usuario autenticado, se le mostrará la opción de autenticarse para acceder a las opciones según sus privilegios dentro de la aplicación, se muestran también la cantidad de usuarios autenticados y no autenticados en línea, y una lista con los nuevos usuarios del sitio.

#### **4.3.2. Concepción general de la ayuda.**

Es necesario para toda aplicación, contar con una ayuda que les permita a sus usuarios consultar cómo funciona cada una de las opciones que brinda la herramienta.

Para los usuarios del sistema, en la página principal del Sitio de Soporte de Software se muestran todas las opciones que se brindan, y una descripción de cada una de ellas que incluye la manera de utilizarlas, lo cual permite conocer todos los servicios que se pueden recibir según el rol de cada usuario. Para el administrador del sistema el Drupal incluye una descripción de la acción a realizar a cada momento. Como una opción importante para brindar ayuda a los usuarios, están los Foros de Discusión, donde se puede consultar con otros usuarios o con el administrador del sistema. Al ser esta aplicación un producto más de los desarrollados por la Dirección de Informatización y utilizado por todos los usuarios de la misma, se podría solicitar soporte para esta y recibir información en cuanto a la manera de usarla.

### **4.3.3. Tratamiento de excepciones.**

Cada formulario creado se encarga de la validación de sus datos para evitar errores de concepto.

Se utilizan mensajes de confirmación, para acciones que son irreversibles como es el caso de las eliminaciones.

La parte de la aplicación, configurada o que funciona a partir de los módulos centrales de Drupal está validada mediante la captura de excepciones del propio Drupal.

### **4.4. Estándares de codificación.**

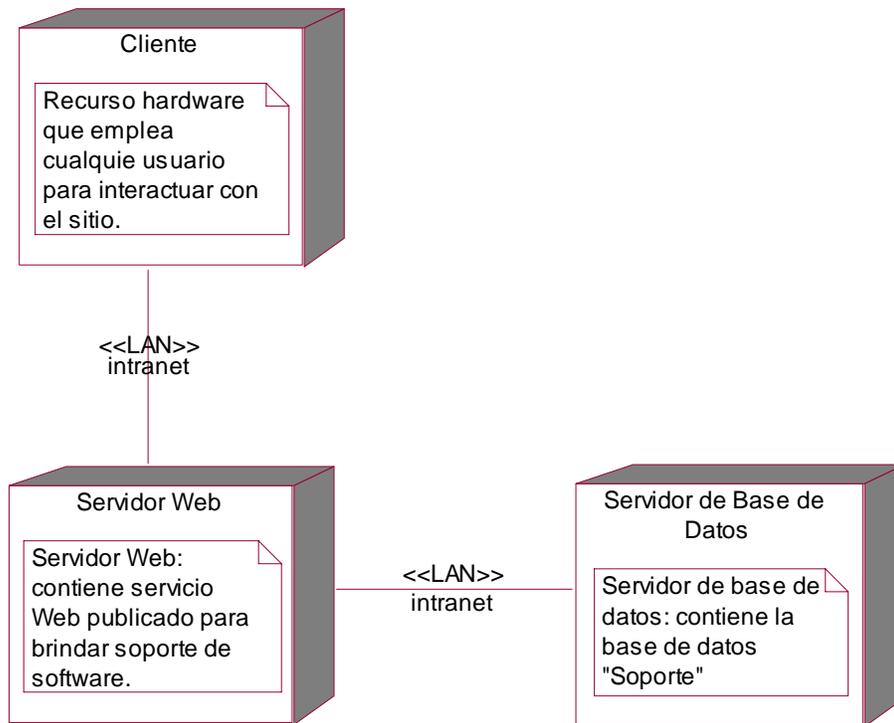
Para la codificación del sistema que se propone se utilizaron los estándares empleados por el CMS Drupal, por ejemplo:

Tags de PHP: fueron usados los tags `<?php ?>` para la definición de código php.

Bloques de instrucciones: el código de las sentencias dentro del bloque se indenta con un tab, para permitir mayor legibilidad. El corchete de apertura “{” se coloca en la misma línea que la instrucción y se alinea el corchete de cierre “}” con la instrucción.

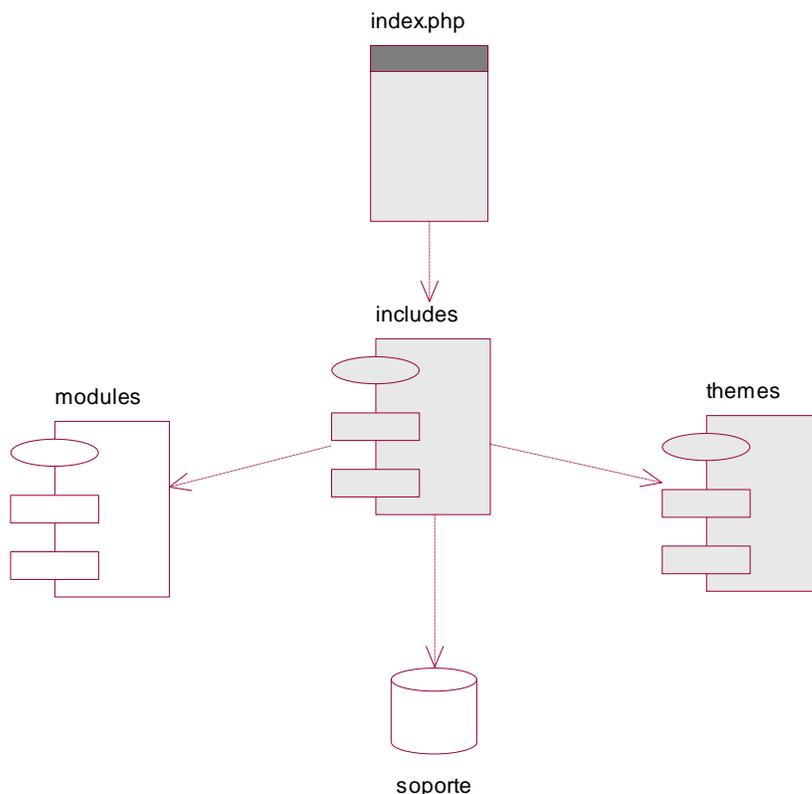
Convenciones de nombres: los nombres de las variables, constantes, clases y funciones usadas se nombran con las letras minúsculas, en caso de más de una palabra para el nombre se separarán estas por un guión bajo.

### 4.5. Diagrama de despliegue.



## 4.6. Diagramas de componentes.

### 4.6.1. Diagrama de componentes del sistema.



Mediante los diagramas de componentes se representan los componentes lógicos de la aplicación así como las relaciones de dependencia entre ellos.

En el curso de este capítulo, han sido objeto de modelación los módulos implementados para las funcionalidades agregadas a las ofrecidas por Drupal. Se reitera que en este epígrafe se hará de igual forma.

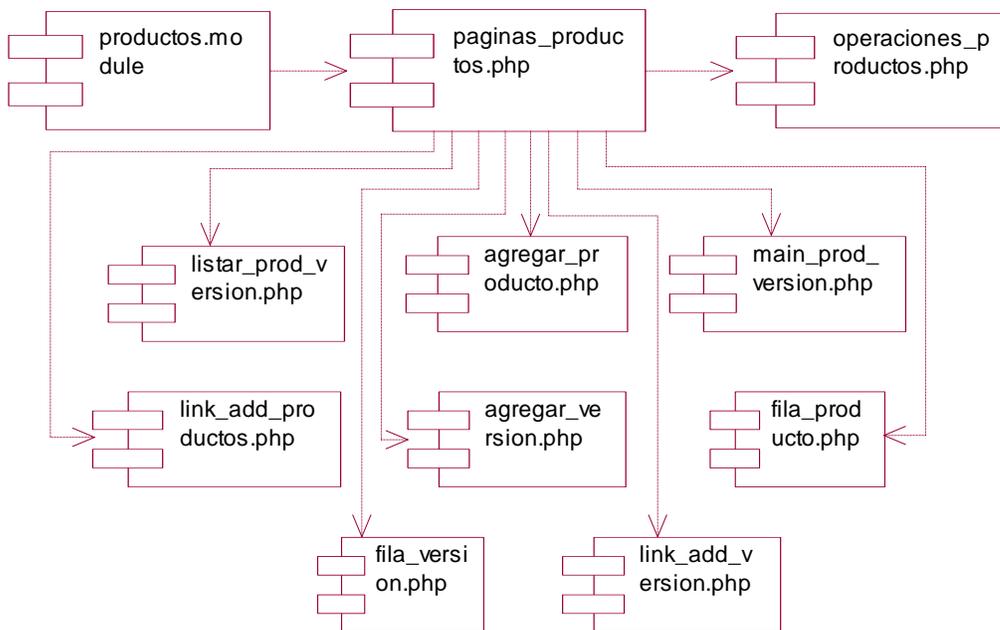
Por la estructura de un módulo implementado para Drupal, existen dos elementos similares para cada uno de los siguientes diagramas (*X.module*, *paginas\_X.php*) y por la necesidad de dar funcionalidad a estos módulos, un tercer elemento similar en cada uno de ellos (*operaciones\_X.php*). Interpretétese X como el nombre del módulo.

- *X.module*: Fichero que contiene la estructuración del módulo y al que Drupal reconoce como tal.

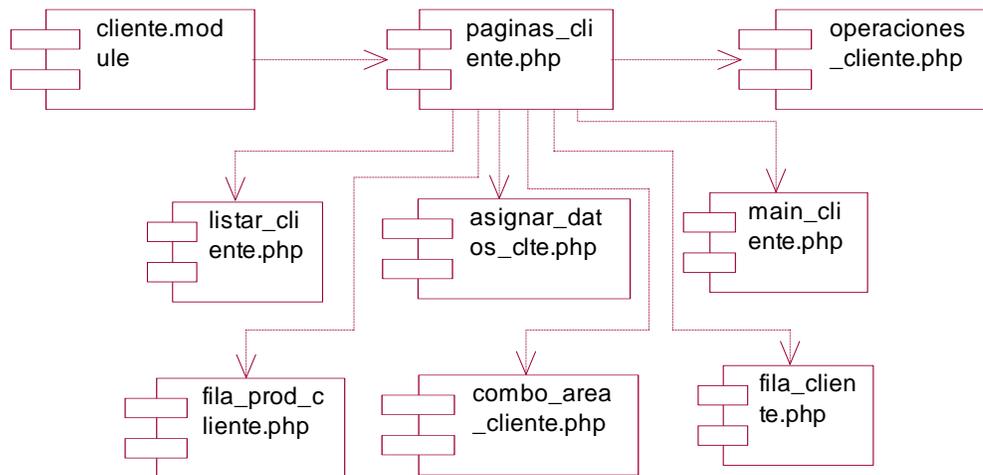
- paginas\_X.php: Fichero que contiene métodos para el trabajo con código HTML.
- operaciones\_X.php: Fichero que contiene la lógica de negocio para el módulo en cuestión.

El resto de los elementos responden a ficheros.php que contienen código HTML, utilizado para la conformación de la página que se ha de mostrar.

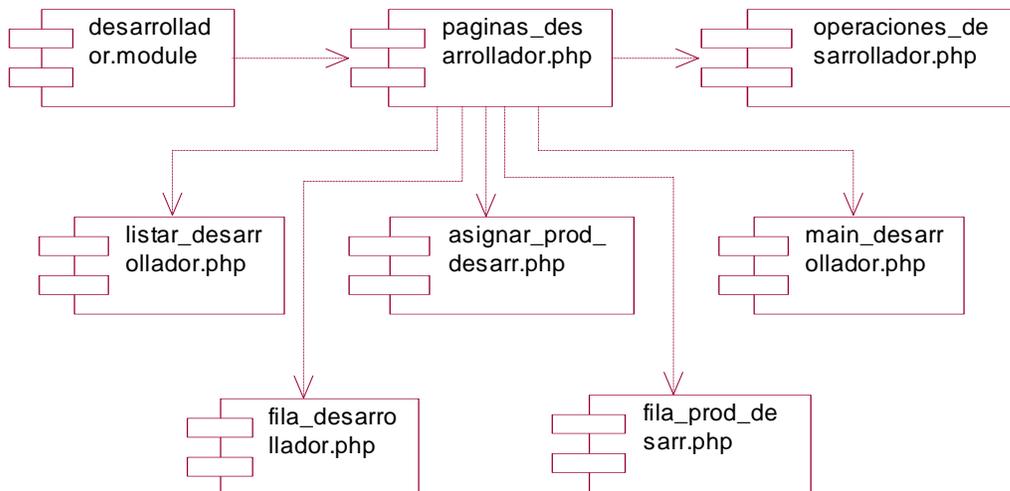
#### 4.6.1.1. Diagrama de componentes del paquete productos.



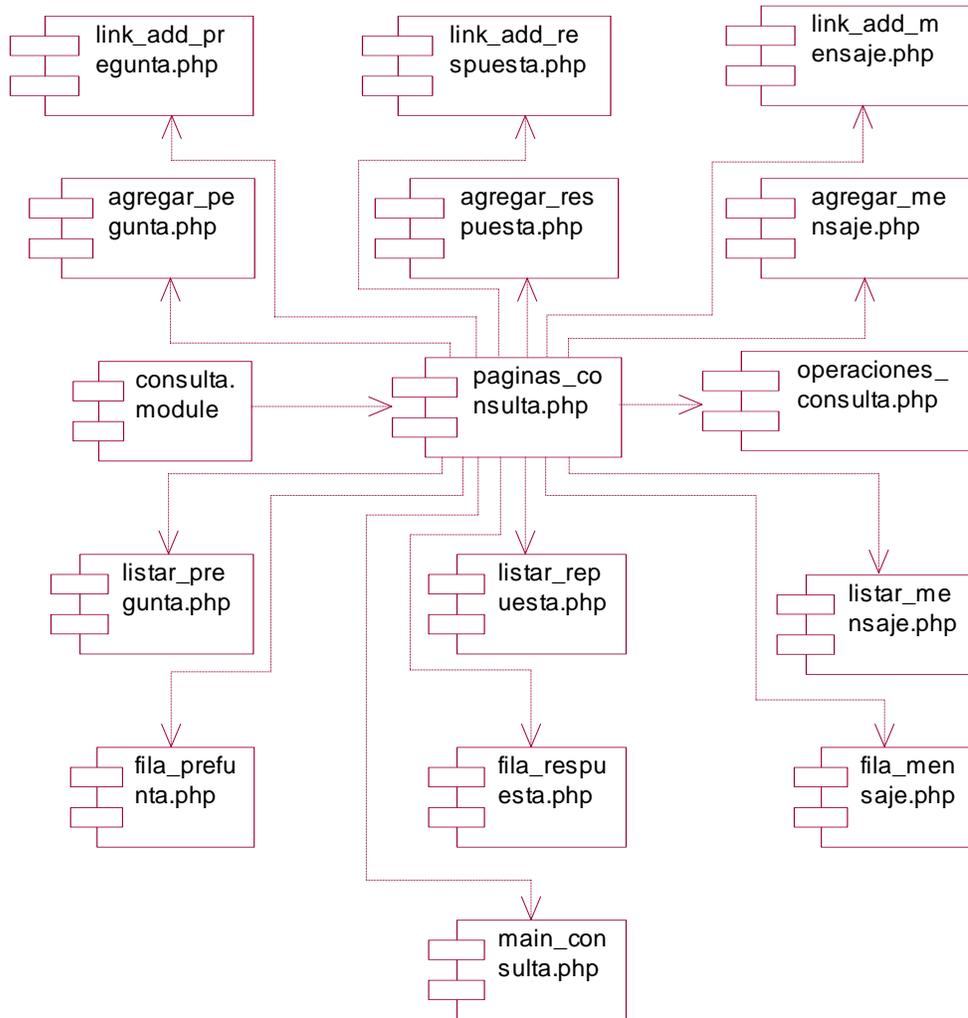
### 4.6.1.2. Diagrama de componentes del paquete clientes.



### 4.6.1.3. Diagrama de componentes del paquete desarrolladores.



#### 4.6.1.4. Diagrama de componentes del paquete consultas.



#### 4.7. Conclusiones.

Se concluye la etapa de diseño del sistema, donde se presentan los diagramas de clases por paquetes de acuerdo a los módulos desarrollados en la aplicación. Se muestra además el diseño de la Base de Datos y el diagrama de clases persistentes. Se plantean los principios de diseño utilizados en cuanto a la interfaz de la aplicación y el tratamiento de excepciones. Finalmente se muestran los estándares de codificación y los diagramas de despliegue y el de componentes. Todo este proceso contribuye a una mejor codificación a la hora de la implementación del sistema.

## Capítulo 5. Estudio de factibilidad.

---

El estudio de factibilidad, beneficios y costo del sistema propuesto, se desarrolla en el presente capítulo aplicando la herramienta COCOMO (CONstructive COSt MOdel),

Este modelo trata de estimar, de una manera rápida en la mayoría de proyectos pequeños y medianos, qué tiempo se requerirá para su realización, cuanto dinero y cuántas personas serán necesarias para su ejecución, también otros indicadores, tales como, cantidad de líneas de código o instrucciones contadas en miles.

### 5.1. Planificación.

➤ Características del proyecto.

- Entradas Externas

Nombre de la entrada externa	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación (Simple, Media y compleja)
Insertar _ producto	1	2	Simple
Insertar _ versión	1	4	Simple
Asignar _ producto	1	2	Simple
Asignar _ versión	1	2	Simple
Asignar _ área	1	1	Simple
Asignar _ pregunta_E	1	2	Simple
Insertar _ pregunta	2	6	Simple
Insertar _ respuesta	2	5	Simple
Insertar _ mensaje	1	5	Simple
Modificar _ producto	1	2	Simple
Modificar _ versión	1	4	Simple

- Salidas Externas

Nombre de la salida externa	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación(Simple, Media y compleja)
Mostrar _ clientes	2	3	Simple
Mostrar _ desarrolladores	1	2	Simple
Mostrar _ productos	1	2	Simple
Mostrar _ versiones	1	3	Simple
Mostrar _ áreas	1	2	Simple
Mostrar _ especialistas	1	3	Simple
Mostrar _ receptores	1	3	Simple

- Peticiones

Nombre de la petición	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación(Simple, Media y compleja)
Mostrar _ preguntas	1	2	Simple
Mostrar _ respuestas	1	1	Simple
Mostrar _ mensajes	1	1	Simple

- Ficheros internos

Nombre del fichero interno	Cantidad de records	Cantidad de Elementos de datos	Clasificación(Simple, Media y compleja)
Área	1	1	Simple
Cliente	1	1	Simple
Trabajador	1	1	Simple
Mensaje	1	5	Simple
Cliente _ producto	1	2	Simple
Inquietud	1	4	Simple

Producto	1	3	Simple
Versión	1	5	Simple
Desarrollador	1	1	Simple
Respuesta_inquietud	1	4	Simple
Respuesta	1	2	Simple
Especialista	1	1	Simple
Receptor	1	1	Simple
Trabajador _directo	1	1	Simple
Pregunta	1	3	Simple
Debate	1	2	Simple

- Puntos de Función desajustados

Elementos	Simples	X Peso	Medios	X Peso	Complejos	X Peso	Subtotal de puntos de función
Ficheros lógicos internos	16	7					112
Entradas externas	11	3					33
Salidas externas	7	4					28
Peticiones	3	3					9
<b>Total</b>							<b>182</b>

- Cantidad de instrucciones fuentes estimadas

Ratio según lenguaje

SQL = 39

PHP = 21

- Para SQL

$UFP = (182 \cdot 10) / 100 = 18.2$

$SLOC = 18.2 \cdot 39 = 709.8$

Líneas de código fuente = 0.709 KSLOC

- Para PHP

$UFP = (182 \cdot 90) / 100 = 163.8$

$SLOC = 163.8 \cdot 21 = 3439.8$

Líneas de código fuente = 3.439 KSLOC

Características	Valor
-----------------	-------

Puntos de función desajustados	182
Lenguaje PHP	21(90%)
Lenguaje SQL	39 (10%)
Instrucciones fuentes por puntos de función (PHP)	3439.8
Instrucciones fuentes (PHP)	3.439
Instrucciones fuentes por puntos de función (SQL)	709.8
Instrucciones fuentes (SQL)	0.709
Instrucciones fuentes(Total)	4,149

### 5.2. Costos.

- Cálculo de las instrucciones fuentes, esfuerzo (PM), tiempo de desarrollo (TDEV), cantidad de hombres y costo.

$$PM_{NS} = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$TDEV_{NS} = C \times (PM_{NS})^F$$

where  $E = B + 0.01 \times \sum_{j=1}^5 SF_j$

where  $F = D + 0.2 \times 0.01 \times \sum_{j=1}^5 SF_j$   
 $= D + 0.2 \times (E - B)$

Cálculo de:	Valor	Justificación
Esfuerzo ( PM )	14.8	<b>A</b> = 2.94, <b>B</b> = 0.91, <b>C</b> = 3.67, <b>D</b> = 0.28 <b>Size</b> : Tamaño estimado (KSLOC).=4.149 <b>SFj : Factores de Escala</b> <b>PREC</b> : Precedencia. = 3.72 <b>FLEX</b> : Flexibilidad. = 2.03 <b>RESL</b> : Riesgos. = 2.83 <b>TEAM</b> : Cohesión del Equipo. = 3.29 <b>PMAT</b> : Madurez de las Capacidades. = 4.68 $\sum SF_j = 16.55$ $E = 0.91 + 0.01 * 16.55 = 1.0755$ $PEM = 1.09$ $PM = 2.94 * 4.149^{1.0755} * 1.09 = 14.8$
Tiempo de desarrollo (TDEV)	(8.5) 9 meses	<b>A</b> = 2.94, <b>B</b> = 0.91, <b>C</b> = 3.67, <b>D</b> = 0.28 $F = 0.28 + 0.2 * 0.01 * 16.55 = 0.313$ <b>TDEV</b> = 3.67 (14.8) <sup>0,313</sup>
Cantidad de personas	(2.4) 2 hombres	14.8 / 6

Costo	\$9235.2	$C = CHM * PM * S$ $C = 2 * 312 * 14.8$
Salario medio	\$312.00	Salario medio del país actualmente.
<b>Multiplicadores de esfuerzo EM</b>		
RCPX	1.00	RELY = NOMINAL DOCU = BAJO CPLX = NOMINAL DATA = NOMINAL
RUSE	1.00	RUSE = 1.00
PDIF	1.00	TIME = NOMINAL STOR = NOMINAL PVOL = NOMINAL
PREX	1.33	APEX=BAJO PLEX= BAJO LTEX= MUY BAJO
FCIL	0.87	TOOL = ALTO SITE = ALTO.
SCED	1.14	SCED = BAJO
PERS	0.83	ACAP = NOMINAL---55% PCAP = NOMINAL---55% PCON = ALTO

### 5.3. Beneficios tangibles e intangibles.

- Beneficios intangibles.

El sistema permitirá brindarle información adicional al cliente, posibilitará la comunicación entre los clientes y la organización desarrolladora de software, brindará el servicio de Soporte de Software todo el tiempo, lo que representa una comodidad para el cliente; pues no tiene que moverse de su puesto de trabajo.

Facilitará el trabajo del grupo de soporte, evitará que el Ingeniero Informático emplee tiempo en el Soporte de Software, lográndose mayor control y organización del trabajo.

La publicación del sitio en una Intranet o el Internet posibilita que cualquier usuario navegue por él, y tenga acceso a la información publicada.

El grupo de Soporte de Software será beneficiado, ya que su trabajo será disminuido debido a la automatización del servicio de Soporte de Software.

- Beneficios tangibles.

Permitirá aumentar la comunicación entre los que brinden el servicio de Soporte de Software a través del sistema y quienes lo reciban, permitiendo el ahorro de recursos de transportación, teléfono y tiempo.

#### **5.4. Análisis de costos y beneficios.**

Los beneficios tangibles e intangibles que posee un producto informático justifican el costo al que va asociado el mismo. Así, un análisis de la relación costo beneficio es importante para decidir la confección.

En el caso de la herramienta propuesta, es factible su implementación, debido a que no existe en la UCI un software con las características de esta, además de que en el país esta esfera puede clasificarse como incipiente, debido a que el desarrollo de la producción de software ha comenzado desde hace pocos años. Este servicio es útil para los grupos y empresas desarrolladoras de software, ya que facilita la comunicación entre los clientes y la empresa e incrementa la calidad de los productos y la satisfacción del cliente. No es necesaria una inversión en los medios técnicos, ni en requerimientos de lenguajes como PHP y MySQL para su ejecución, además, se ha estimado que el costo por concepto de salario es aceptable así como el tiempo de ejecución.

#### **5.5. Conclusiones.**

Se concluye el análisis de la factibilidad económica del sistema propuesto. Se muestra la planificación realizada y se realizaron los cálculos correspondientes al costo del producto, teniendo en cuenta los factores que influyen en este parámetro, tiempo de ejecución, cantidad de desarrolladores, salario medio, entre otros. Se plantean los beneficios tangibles e intangibles que posee el sistema y finalmente se realiza un análisis costo-beneficio. Con todo este proceso se concluye que el sistema es factible en su implementación.

## ***Conclusiones.***

---

El soporte de software surge como una necesidad de las entidades productoras de brindarle al cliente confianza en sus productos de software. En el proceso de soporte se brindan respuestas a las inquietudes del cliente y se puede llegar a la modificación de las características de los productos desarrollados para satisfacer las necesidades de los clientes.

Se propone un sistema basado en el Web para el soporte de software el cual incluye las opciones básicas de los sitios de soporte de software, contando con un seguimiento de incidencias para mantener informados a los usuarios del estado en que se encuentra el procesamiento de su inquietud. El sistema además permite calcular estadísticas de las incidencias por grupos de resolución, tiempos de resolución, usuarios y otros parámetros, estas tienen como objetivo la realización de un análisis de la actividad del grupo, conllevando al mejoramiento del servicio y la operabilidad de los usuarios.

Finalmente se arriba a las siguientes conclusiones:

- El servicio de Soporte de Software es importante para garantizar la calidad del software y la satisfacción del cliente, además es un factor de gran influencia en el buen desempeño de las entidades clientes, coadyuvando a la retención de los mismos.
- Se propone un sistema de soporte de software utilizando el Web, el cual cuenta con las opciones de los sitios de soporte necesarias para garantizar un servicio de calidad y con rapidez
- Para la construcción de la herramienta se determinó utilizar el CMS Drupal que implementa con el lenguaje de programación PHP, como Sistema Gestor de Base de Datos el MySQL y para la modelación en la etapa de Análisis y Diseño se utilizó RUP.

A través del sistema propuesto se brinda el servicio de Soporte de Software a los usuarios de la Dirección de Informatización de la UCI, por lo que reporta entre otros los siguientes beneficios:

- Posibilita la comunicación entre los clientes y de estos con la Dirección.
- Comodidad para el cliente.
- Facilitar el trabajo, logrando un mayor control y organización.

Con el estudio realizado y el sistema que se ha implementado, se cumple con el objetivo propuesto: desarrollar un sistema integral para el soporte de software para el Grupo de Soporte de Software de la Dirección de Informatización de la Universidad de las Ciencias Informáticas.

## ***Recomendaciones.***

---

Para la continuidad del desarrollo de este trabajo se recomienda:

- Permitir la búsqueda en la base de preguntas y respuestas, utilizando métodos más eficientes. Sugiriendo utilizar técnicas de minería de textos, que permitan estimar la proximidad de una pregunta realizada por el usuario a todas las que se encuentren almacenadas.
- Añadir una funcionalidad de estimar la satisfacción del cliente con los productos y con el servicio de Soporte de Software, utilizando la propia herramienta.

## Bibliografía.

---

- [ARZ02]. Arza Pérez, L., Ortiz Concepción, A. *SOSoft. Herramienta para el Soporte de Software*. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, junio 2002.
- [BEI05]. Beisse, Fred. *A Guide to Computer User Support for Help Desk and Support Specialists*  
Chapter 6: Help Desk Operation  
<http://apollo.saultc.on.ca/~fturco/courses/csa2o1/fall2002/chap06.htm>
- [CON05]. Convergente SLA. *Servicio a Soporte*. 2005. (11/05/2005)  
[http://www.convergence-i.com/COL/Soporte/Soporte\\_SLA.htm](http://www.convergence-i.com/COL/Soporte/Soporte_SLA.htm)
- [CRU97]. Cruz Bovea, Cesar. *Cómo implementar un sistema de gestión de la calidad en su empresa; Control de los productos no conformes*. 1997 <http://www.monografias.com/trabajos22/productos-no-conformes/productos-no-conformes.shtml>
- [DON02]. Dondo, Agustín. *¿Por qué elegir PHP?* 2002. (12/05/2005)  
<http://www.programacion.net/php/articulo/porquephp>
- [EST02]. *Estrategia para el uso del Software Libre en Cuba*. 2002. (12/04/2005)  
[www.linux.cu](http://www.linux.cu)
- [FEA05]. *Feature overview*. 2005. (11/05/2005)  
<http://drupal.org/features>
- [FER95]. Fernández Carrasco, Oscar M., García León, Delba. Beltrán Benavides, Alfa. *Un enfoque actual sobre la calidad del software*. *ACIMED* 3(3):40-42, septiembre-diciembre, 1995. (4/04/2005)  
[http://www.bvs.sld.cu/revistas/aci/vol3\\_3\\_95/aci06395.htm](http://www.bvs.sld.cu/revistas/aci/vol3_3_95/aci06395.htm)
- [GAR03]. García Puente, Carlos Ignacio. *Software de Libre Disponibilidad. Su utilización en el Estado*. 2003. (12/05/2005)  
<http://www.monografias.com/trabajos18/software-libre-estado/software-libre-estado>
- [GNU05]. GNU. Wikipedia. *La enciclopedia libre*. 2005. (11/05/2005)  
<http://es.wikipedia.org/wiki/GNU>
- [GON97]. González, Carlos. *Conceptos generales de calidad total*. 1997.  
<http://www.monografias.com/trabajos11/conge/conge.shtml>
- [HIS03]. *History of PHP and related projects*. 2003. (11/05/2005)  
<http://www.php.net/history>
- [JAC00]. Jacobson, I.; Booch, G. y Rumbaugh, J.; *El Proceso Unificado de Desarrollo de software*. 2000. Addison-Wesley.
- [OPE05]. *OpensourceCMS*, 2005. (12/05/2005)  
<http://www.opensourcecms.com>
- [SAN02]. Sánchez, Eduardo. *¿Qué es MySQL?* 2002. (11/05/2005)  
[http://www.mysql.com/products/what\\_is\\_mysql.html](http://www.mysql.com/products/what_is_mysql.html)

- [SIS03] *Sistemas de manejo de contenidos (CMS)*. 2003.  
<http://observatorio.cnice.mec.es/index.php?module=subjects&func=viewpage&pageid=1>
- [SIS05]. *Sistema operativo GNU - Fundación para el software libre*. 2005.  
(11/05/2005) <http://www.gnu.org/home.es.HTML>
- [ROB02]. Robertson, J., *How to evaluate a content management system*.  
2002. (12/04/2005)  
[http://www.steptwo.com.au/papers/kmc\\_evaluate/index.HTML](http://www.steptwo.com.au/papers/kmc_evaluate/index.HTML)
- [ROB03]. Robertson, J., *Looking towards the future of CM*. 2003.  
(12/04/2005)  
[http://www.steptwo.com.au/papers/cmb\\_future/index.HTML](http://www.steptwo.com.au/papers/cmb_future/index.HTML)
- [VIL01a]. Villate, Javier. *Software libre y regulación*. 2001 (11/05/2005)  
<http://www.cibersociedad.net/archivo/articulo.php?art=39>
- [VIL01b]. Villate, Javier. *La propiedad intelectual en la nueva era digital*.  
2001. (11/05/2005)  
<http://www.cibersociedad.net/archivo/articulo.php?art=40>
- [VAL05] Valle, Amaury E. *Impulsará Cuba el sistema operativo Linux*.  
2005. (12/04/2005)  
La Ventana. Portal Informativo de la Casa de las Américas  
<http://laventana.casa.cult.cu>

## ***Glosario de términos.***

---

**Apache** - servidor de páginas Web de código abierto para diferentes plataformas (UNIX, Windows, etc.)

**Blog (Web log)** - Diario en formato Web. Puede ser un diario personal o un conjunto de noticias, ordenado por fecha.

**CMS** (Content Management System) - Sistema que facilita la gestión de contenidos en todos sus aspectos: creación, mantenimiento, publicación y presentación. También se conoce como Web Content Management (WCM) sistema de gestión de contenido de webs.

**CSS** (Cascading Style Sheets) - Las hojas de estilo en cascada contienen un conjunto de etiquetas que definen el formato que se aplicará al contenido de las páginas de una Web. Se llaman "en cascada" porque una hoja puede heredar los formatos definidos en otra hoja de forma que no hace falta que vuelva a definirlos. Estas hojas permiten la separación entre el contenido y la presentación en una Web.

**GPL** (General Public License) - Licencia que permite el uso y modificación del código para desarrollar software libre, pero no propietario.

**HTML** (HyperText Markup Language) - Lenguaje basado en marcas que indican las características del texto, utilizado para definir documentos de hipertexto en webs.

**HTTP** (HyperText Transfer Protocol) - Protocolo cliente-servidor utilizado para el intercambio de páginas Web (HTML)

**LGPL** (Lesser General Public License) - Licencia que permite el uso y modificación de librerías de código para desarrollar software libre o propietario. Antes conocida como Library GPL.

**PHP** (PHP Hypertext Preprocessor) - Lenguaje de programación para el desarrollo de webs dinámicas, con sintaxis parecida a la C. Originalmente se conocía como Personal Home Page tools, herramientas para páginas personales (en Internet).

**URL** (Uniform Resource Locator) - Dirección de un recurso en la Web. Tiene el formato `protocol://màquina.domini:port/ruta/recurs`. Por ejemplo `http://www.uoc.edu/dt/20396/index.HTML` donde no se indica el puerto porque el protocolo HTTP tiene uno por defecto (80).

**Web** - sistema para presentar información en Internet basado en hipertexto. Cuando se utiliza en masculino (el Web, un Web) se refiere a un sitio Web entero, en cambio si se utiliza en femenino (la Web, una Web) se refiere a una página Web concreta dentro del sitio Web.

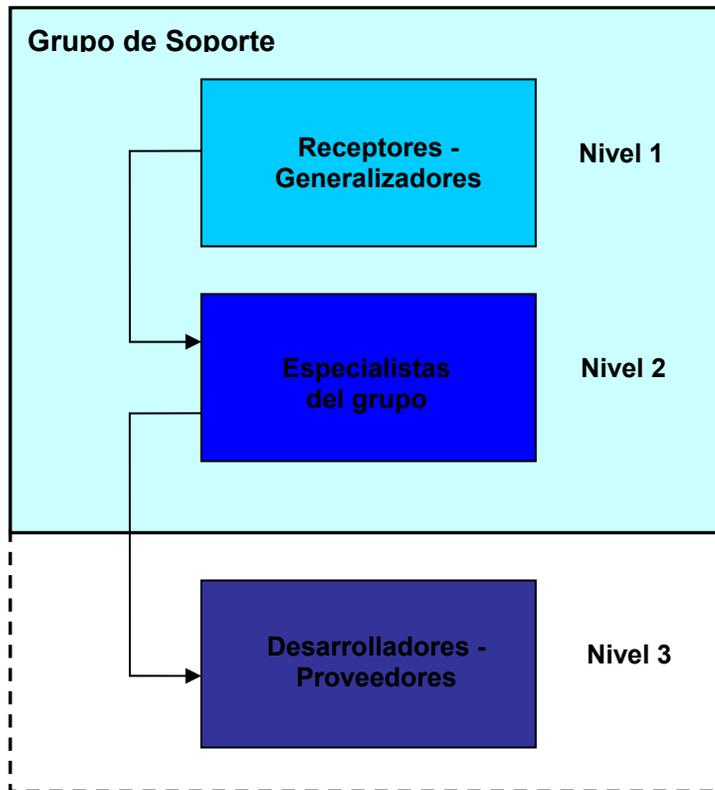
**Web log** - Ver 'Blog'.

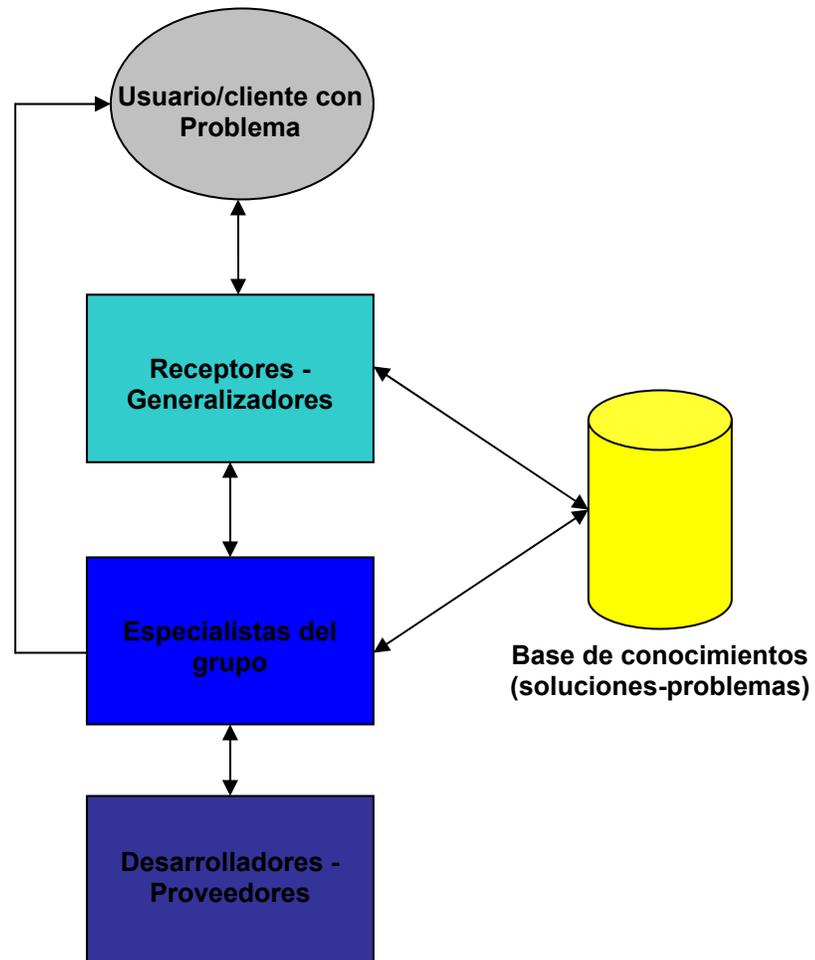
**WYSIWYG** (What You See Is What You Get) - Traducido: lo que ves es lo que obtienes, que aplicado a la edición significa trabajar con un documento con el aspecto real que tendrá. Editar una página de HTML en un editor que no sea WYSIWYG, implica trabajar con los códigos que indican el formato que tendrá el texto, sin ver el resultado final.

## Anexos:

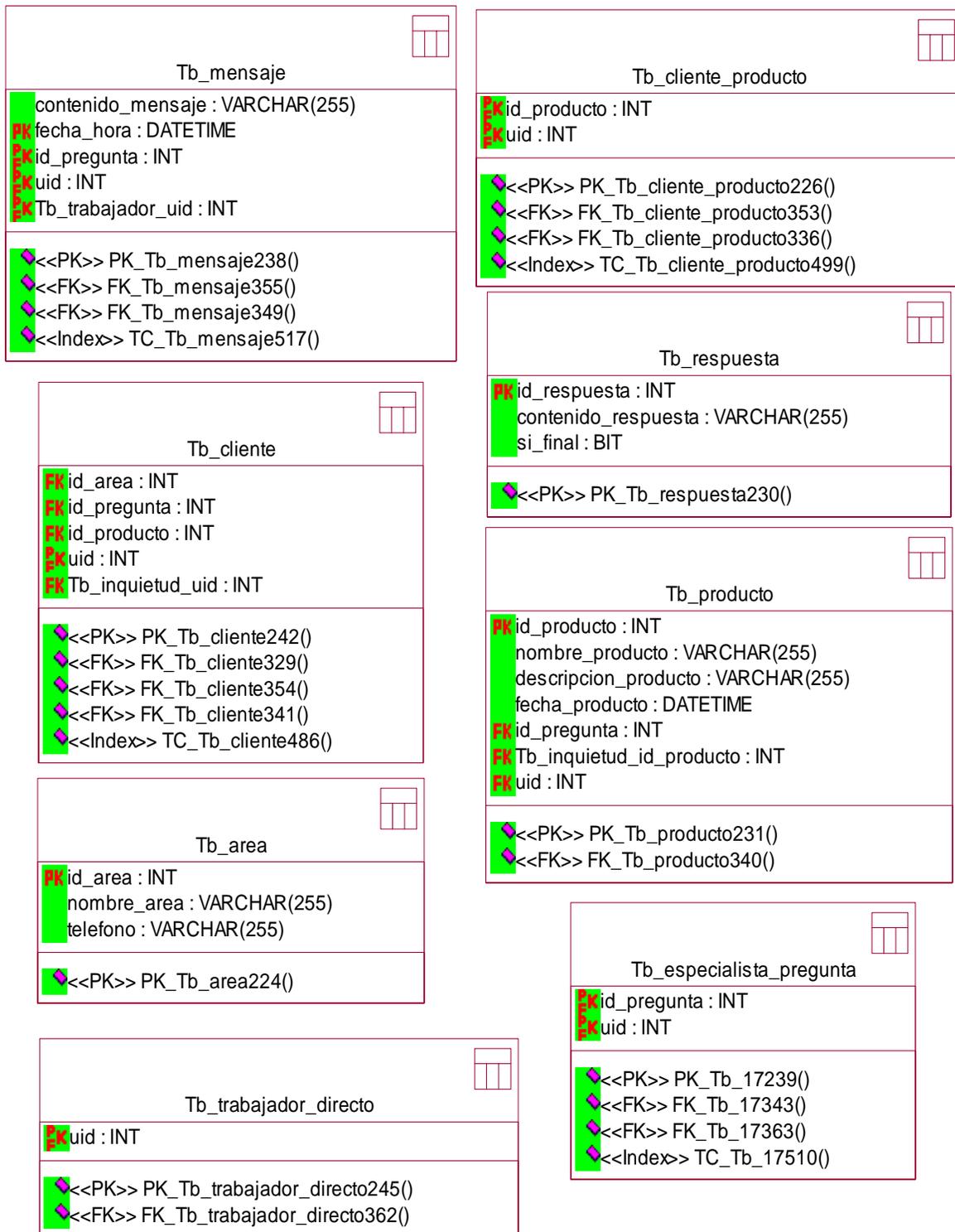
---

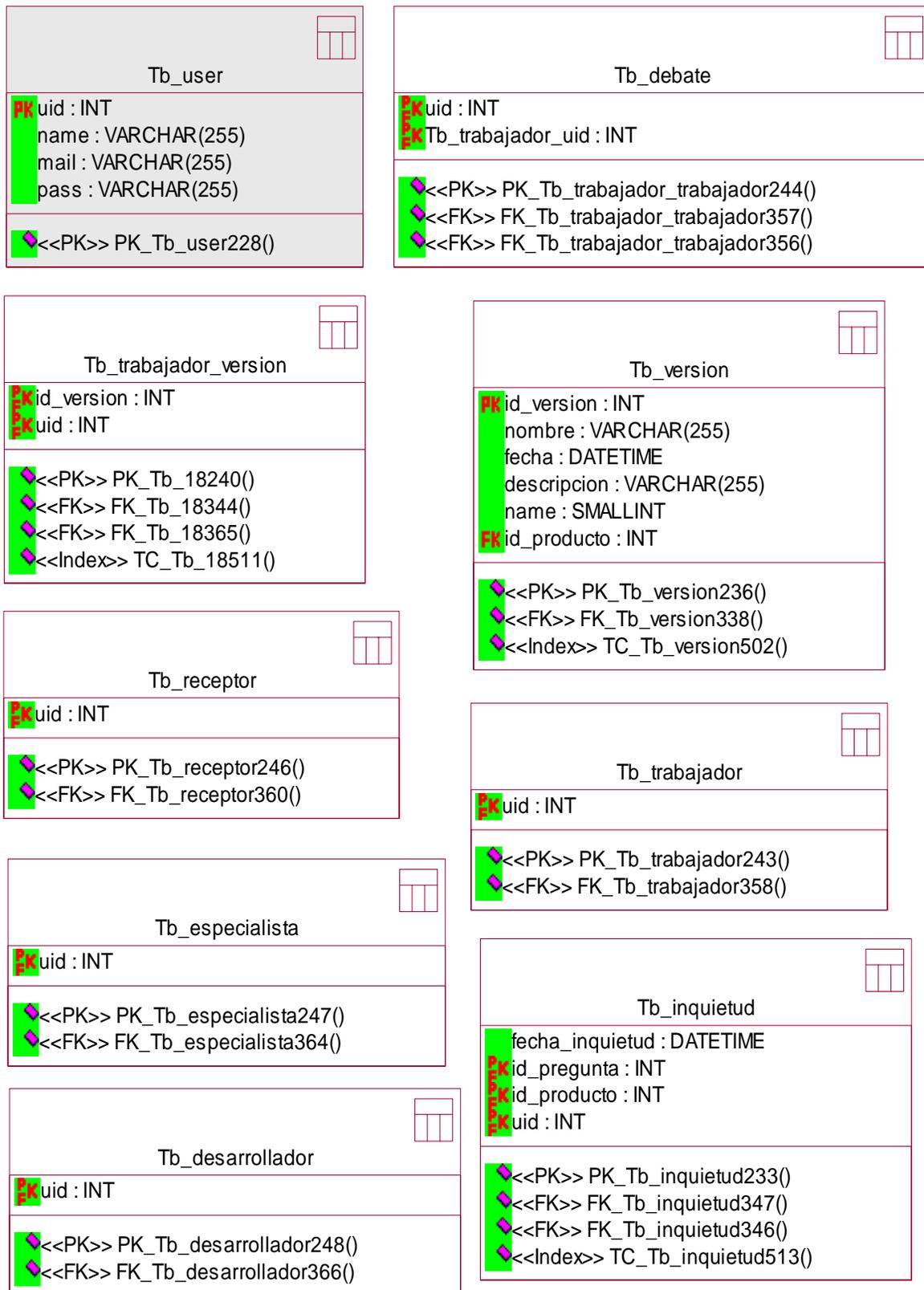
### Anexo 1. Estructura global del grupo de soporte de la UCI.



**Anexo 2. Flujo de trabajo del grupo de soporte de la UCI.**

### Anexo 3. Tablas del modelo de datos.





Tb_respuesta_inquietud	
fecha_respuesta : DATETIME	
<b>PK</b> Tb_respuesta_inquietud_ID : INT	
<b>FK</b> id_respuesta : INT	
<b>FK</b> id_pregunta : INT	
<b>FK</b> uid : INT	
<b>&lt;&lt;PK&gt;&gt;</b> PK_Tb_respuesta_inquietud223()	
<b>&lt;&lt;FK&gt;&gt;</b> FK_Tb_respuesta_inquietud361()	
<b>&lt;&lt;FK&gt;&gt;</b> FK_Tb_respuesta_inquietud348()	
<b>&lt;&lt;FK&gt;&gt;</b> FK_Tb_respuesta_inquietud335()	
<b>&lt;&lt;Index&gt;&gt;</b> TC_Tb_respuesta_inquietud498()	
<b>&lt;&lt;Index&gt;&gt;</b> TC_Tb_respuesta_inquietud516()	

Tb_pregunta	
<b>PK</b> id_pregunta : INT	
contenido_pregunta : VARCHAR(255)	
fecha_pregunta : DATETIME	
si_atendida : BIT	
si_respondida : BIT	
<b>FK</b> Tb_inquietud_id_pregunta : INT	
<b>FK</b> id_producto : INT	
<b>FK</b> uid : INT	
<b>FK</b> Tb_receptor_uid : INT	
<b>&lt;&lt;PK&gt;&gt;</b> PK_Tb_pregunta237()	
<b>&lt;&lt;FK&gt;&gt;</b> FK_Tb_pregunta359()	
<b>&lt;&lt;FK&gt;&gt;</b> FK_Tb_pregunta339()	