

**Universidad de las Ciencias Informáticas
Facultad 1**



Título: Sistema de Ayuda para la edición Nova Escritorio.

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas.

Autores: Yoency Leyva Ochoa
José Gustavo Suárez Matilla

Tutores: Ing. Yaima Oval Riverón
Ing. Susana Sánchez Ortiz

Co-Tutor: Ing. Alfredo Pérez Benitez

**La Habana
Junio 2012**

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Yoency Leyva Ochoa

Firma del Autor

José Gustavo Suárez Matilla

Firma del Autor

Ing. Yaima Oval Riverón

Firma de la Tutora

Ing. Susana Sánchez Ortiz

Firma de la Tutora

Ing. Alfredo Pérez Benitez

Firma del Co-Tutor

Frase



“Si los jóvenes fallan, todo fallará. Es mi más profunda convicción que la juventud cubana luchará por impedirlo. Creo en ustedes.”

Fidel Castro Ruz

Resumen

En la actualidad existen sistemas inteligentes que permiten un mejor control de los procesos de la vida diaria, estos sistemas han surgido gracias al avance de la Inteligencia Artificial. La distribución cubana GNU/Linux Nova carece de un mecanismo para proporcionar ayuda a sus usuarios ante los problemas que puedan presentar mediante el uso de la distribución. Por ello en el presente trabajo se realizó un estudio de los diferentes tipos de sistemas de ayuda, sus características, clasificaciones; así como, lenguajes y herramientas para su implementación.

Ante la necesidad de ayudar al usuario, se obtuvo como resultado práctico de la investigación un sistema de ayuda para la edición Nova Escritorio. Este sistema utiliza características de los Sistemas Basados en el Conocimiento (SBC), es del tipo Sistemas de Ayuda Inteligentes (pasivo), nombrado “**Ayuda y Soporte**” y permite identificar mediante preguntas los problemas más frecuentes con los que se enfrenta el usuario en su trabajo diario. Para su diseño e implementación se utilizó tecnologías modernas acorde a la importancia que posee brindar un sistema actualizado. Para asegurar el correcto funcionamiento y calidad del *software* se realizó la verificación del cumplimiento de las especificaciones, de los mecanismos de inferencia y de la Base de Conocimientos, además se aplicaron pruebas Unitarias y de Aceptación.

Palabras clave: *ayuda, Nova, sistema de ayuda, sistemas inteligentes.*

Índice

DECLARACIÓN DE AUTORÍA	I
FRASE	II
RESUMEN	III
INTRODUCCIÓN	1
CAPÍTULO 1: “LOS SISTEMAS DE AYUDA PARA SISTEMAS OPERATIVOS”	5
1.1 SISTEMAS DE AYUDA PARA ENTORNOS INFORMÁTICOS.....	5
1.2 CARACTERÍSTICAS DE LOS SISTEMAS DE AYUDA PARA ENTORNOS INFORMÁTICOS.....	6
1.3 CLASIFICACIÓN DE LOS SISTEMAS DE AYUDA.....	6
1.4 EJEMPLOS DE SISTEMAS DE AYUDA PARA ENTORNOS INFORMÁTICOS.....	8
1.5 LOS SISTEMAS DE AYUDA PARA SISTEMAS OPERATIVOS.....	9
1.6 TIPO DE SISTEMA DE AYUDA A UTILIZAR.....	11
1.7 TECNOLOGÍA A UTILIZAR.....	12
1.8 LENGUAJES UTILIZADOS EN LA IMPLEMENTACIÓN DE SISTEMAS DE AYUDA.....	12
1.9 HERRAMIENTAS UTILIZADAS EN LA IMPLEMENTACIÓN DE SISTEMAS DE AYUDA.....	13
1.10 ASPECTOS ARQUITECTÓNICOS DEL SISTEMA DE AYUDA.....	13
1.11 METODOLOGÍA PARA EL DESARROLLO DEL SISTEMA DE AYUDA.....	15
1.12 MODELO PARA DESARROLLAR SISTEMAS DE AYUDA INTELIGENTE.....	16
CAPÍTULO 2: “SISTEMA DE AYUDA PARA LA EDICIÓN NOVA ESCRITORIO”	18
2.1 SISTEMA DE AYUDA PROPUESTO.....	18
2.2 MODELO QUE SE UTILIZA PARA DESARROLLAR EL SISTEMA DE AYUDA.....	20
2.3 PROCESO DE DESARROLLO DEL PROYECTO BASADO EN SXP.....	20
2.4 MODELO DE DOMINIO.....	21
2.5 LISTA DE RESERVA DEL PRODUCTO.....	22
2.6 HISTORIAS DE USUARIO.....	24
2.7 PLAN DE RELEASE.....	25
2.8 DISEÑO CON METÁFORAS.....	25
2.9 DIAGRAMA DE COMPONENTES.....	27
CAPÍTULO 3: “DISEÑO Y REALIZACIÓN DE PRUEBAS”	29
3.1 PRUEBAS DE ACEPTACIÓN.....	29
3.2 RESULTADOS DE LAS PRUEBAS DE ACEPTACIÓN.....	32
3.3 VERIFICACIÓN DEL SISTEMA DE AYUDA INTELIGENTE.....	32
CONCLUSIONES GENERALES	35
RECOMENDACIONES	36
REFERENCIAS BIBLIOGRÁFICAS	37
BIBLIOGRAFÍA CONSULTADA	40
ANEXOS	44
GLOSARIO DE SIGLAS Y TÉRMINOS	71

Introducción

En la actualidad el avance de las Tecnologías de la Información y las Comunicaciones (TIC) está en constante desarrollo, lo que contribuye a la automatización de procesos y tareas que sólo eran posibles realizar por los seres humanos. Este avance ha permitido el surgimiento de la robótica y los sistemas inteligentes en la era de la informática como alternativa para sustituir las funciones del ser humano. Estos avances científicos han sido posibles gracias a la Inteligencia Artificial (IA)¹ que es una rama de la computación, un ejemplo concreto son los Sistemas de Ayuda Inteligentes (SAI), que tienen la particularidad de proporcionar asistencia activa o pasiva a los usuarios de un sistema informático. Los Sistemas de Ayuda Inteligentes son un tipo especial de sistemas de ayuda, que tienen un objetivo doble, por un lado instruir al usuario sobre cómo operar con el *software* a la vez que se promueve una mayor comprensión del funcionamiento del sistema. [1]

Por otro lado, el proceso de migración al *software* libre va tomando fuerzas en el mundo, pues constituye una solución para el desarrollo de la informática enfocada a un cambio de paradigma. El proceso de migración es factible en muchos países por constituir una alternativa al *software* propietario, especialmente para Cuba por ser un país bloqueado económicamente. Ante el actual desarrollo de las Tecnologías de la Información y las Comunicaciones y de los procesos de migración en el mundo, Cuba ha venido incursionando aceleradamente en el campo de las ciencias. Específicamente en la informática desde diferentes ópticas y es uno de los países que trabaja en la filosofía del código abierto (*open source*)². Además, enfrenta el reto de informatizar su sociedad, con el propósito de integrarse plenamente a la infraestructura de la información [2]. Para complementar este objetivo, se ha enfocado en la producción de *software*, y para lograrlo se ha apoyado principalmente en los Centros de desarrollo de *software*, en la empresa encargada de ofrecer Soluciones Integrales en Tecnologías de la Información para la Informatización de la Sociedad Cubana (DESOFIT S.A) y en la Universidad de las Ciencias Informáticas (UCI).

La Universidad de las Ciencias Informáticas tiene como misión principal formar profesionales altamente calificados en la rama de la Informática, donde se vincula el estudio, la producción y la investigación. Está estructurada por facultades, que se dedican a una o varias líneas de producción. En la Facultad 1 radica el Centro de Software Libre (CESOL) encargado de desarrollar la distribución cubana GNU/Linux Nova, además de ejecutar procesos de migración a plataformas de código abierto.

¹Es la ciencia e ingeniería de hacer máquinas inteligentes, especialmente programas de cómputo inteligentes.

²Software distribuido y desarrollado libremente.

El proyecto productivo Nova, cuenta con las líneas de proyectos de soluciones finales o ediciones de producción, Nova Servidores (*Nova 2011-Servidores*), Nova Ligerero (*Nova 2011-Ligerero*) y Nova Escritorio (*Nova 2011-Escritorio*), esta última orientada al usuario que no sea experto, para ser utilizada en entornos laborales, escolares y el hogar. [3]

En lo adelante los autores para referirse a la frase o expresión sistema operativo utilizarán las siglas SO, por ejemplo SO Unix, SO Nova u otra distribución. Su uso responde a que se utiliza en diversas ocasiones, un ejemplo aplicado sería, el SO Nova.

Ante la necesidad de contar con un sistema para brindar asistencia de soporte técnico y gestionar los reportes, incidencias o problemas que presenten los usuarios de Nova, se creó en la UCI la aplicación Web NovaDesk 1.0 [4]. Con el objetivo de incorporarle nuevas funcionalidades, surgió en el 2010 NovaDesk 2.0: entre sus funcionalidades se destacan la mensajería instantánea, notificación vía correo electrónico de la información de un determinado reporte y una sala de *chat* para los usuarios [5]. Para que la aplicación brindara más utilidad y los usuarios tuviesen una vía directa para resolver sus problemas, se le incorporó un Sistema Experto para la gestión de la Base de Conocimientos de NovaDesk [6].

Aunque NovaDesk presta servicios de utilidad, los usuarios no lo utilizan con frecuencia, debido en muchos casos a la demora de las respuestas a sus problemas, tienen que conocer la dirección web de la aplicación y dependen de la conexión de red. Desde hace un tiempo no se utiliza en la universidad y no viene incluida en la imagen ISO³ del sistema operativo Nova. Las limitaciones antes mencionadas provocan que los usuarios en muchas ocasiones no encuentren solución a sus problemas con el uso del SO Nova.

La mayoría de los sistemas operativos existentes cuentan con una vía para darle respuesta a los problemas que presentan sus usuarios. Nova como SO, utiliza la paquetería de la distribución Ubuntu que cuenta con la ayuda de **GNOME** (*Yelp*) encaminada a un fin específico, pero no se acopla a las características, expectativas y temas específicos de Nova. Atendiendo a lo planteado, las líneas de producción de Nova, como es el caso de Nova Escritorio (edición más utilizada durante el proceso de migración que ejecuta el país), tampoco tienen el mecanismo o forma de ayudar a sus usuarios.

El **problema científico** que se presenta es: ¿Cómo lograr que la edición Nova Escritorio le brinde una mejor respuesta a los problemas más frecuentes de sus usuarios?

³Archivo donde se almacena una copia o imagen exacta de un sistema de ficheros.

Se declara como **objeto de estudio** los Sistemas de Ayuda, enmarcado en el **campo de acción** los Sistemas de Ayuda para Sistemas Operativos.

En correspondencia con el problema planteado se propone como **objetivo general**: Desarrollar un Sistema de Ayuda para la edición Nova Escritorio que permita resolver los problemas más frecuentes de sus usuarios.

Derivándose los siguientes **objetivos específicos**:

- Sistematizar los aspectos teóricos relacionados con los Sistemas de Ayuda para Sistemas Operativos.
- Proponer un Sistema de Ayuda para la edición Nova Escritorio.
- Implementar la solución propuesta.
- Realizar las pruebas de aceptación al Sistema de Ayuda implementado.

Para darle cumplimiento a los objetivos trazados se definen las siguientes **tareas**:

- Revisión y análisis de la bibliografía relacionada con los Sistemas de Ayuda.
- Análisis y diseño de la propuesta de un Sistema de Ayuda.
- Implementación del Sistema de Ayuda.
- Realización de las pruebas de aceptación por historias de usuarios.

En la presente investigación, se sostiene como **idea a defender** que el desarrollo de un Sistema de Ayuda le permitirá a la edición Nova Escritorio dar respuesta a los problemas más frecuentes de sus usuarios.

Para darles solución a las tareas planteadas se usaron métodos teóricos y empíricos como métodos científicos de la investigación.

Los **métodos teóricos** permiten estudiar las características del objeto de investigación que no son observables directamente, facilitan la construcción de modelos e hipótesis de investigación. Para su ejecución se apoyan en el proceso de análisis y síntesis. Inductivo-deductivo, histórico-lógico y analítico-sintético: son ejemplos de métodos teóricos. [7]

Por su parte los **métodos empíricos** describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional. Aunque existen diversas opiniones la mayoría de los autores concuerdan que los métodos empíricos generales son: la observación, la medición, la entrevista y la encuesta. [8]

Métodos teóricos:

- Analítico-Sintético, para estudiar diferentes fuentes bibliográficas relacionados con los sistemas de ayuda. Se realizaron además resúmenes y valoraciones de conceptos relevantes relacionados con los sistemas de ayuda para sistemas operativos.
- Histórico-Lógico, para determinar y estudiar los antecedentes históricos relacionados con los sistemas de ayuda.

Métodos empíricos:

- Observación, con el objetivo de analizar y observar las funcionalidades de los sistemas de ayuda de diferentes sistemas operativos.
- Entrevista, aplicada a los expertos del proyecto productivo Nova para conocer las especificaciones del sistema de ayuda.

El contenido del documento está dividido en tres capítulos, las conclusiones, recomendaciones, anexos y un glosario de siglas y términos.

Capítulo 1: “Los Sistemas de Ayuda para Sistemas Operativos”, en este capítulo se realiza el estudio del estado del arte, haciendo referencia a los aspectos fundamentales relacionados con los sistemas de ayuda; como conceptos, características, tipos, ejemplos y tecnologías orientadas a su desarrollo. También, se define la arquitectura del *software*, la metodología y el modelo a aplicar.

Capítulo 2: “Sistema de Ayuda para la edición Nova Escritorio”, en este se hace alusión a la solución propuesta, planteándose cómo se va a desarrollar la misma y se definen las funcionalidades a implementar. Además, se describe el proceso ágil basado en las historias de usuario correspondiente a las funcionalidades.

Capítulo 3: “Diseño y realización de pruebas”, es aquí donde se describen, diseñan, realizan y controlan los casos de pruebas; también se realiza la verificación del Sistema de Ayuda inteligente implementado.

Capítulo 1: “Los Sistemas de Ayuda para Sistemas Operativos”.

1.1 Sistemas de Ayuda para entornos informáticos.

El desarrollo de sistemas de ayuda es un campo relativamente reciente en el diseño de aplicaciones informáticas, no es habitual encontrarlos en programas anteriores al año 1975. Sin embargo, en la actualidad la mayor parte de las aplicaciones incluyen algún tipo de ayuda. En teoría, con la mejora de las interfaces y del diseño de programas, se hace posible construir aplicaciones tan sencillas de utilizar que el usuario no necesite ayuda adicional. Sin embargo, circunstancias como la creciente complejidad de los sistemas informáticos, su continua evolución y su uso generalizado por usuarios no expertos, hacen que los sistemas de ayuda sean un componente fundamental para simplificar el uso y el aprendizaje de los entornos informáticos⁴.

En los últimos años autores como Winkels, Kearsley, Duffy, Boy, Breuker y Baltasar, han publicado diversos trabajos sobre los sistemas de ayuda, lo que indica que se trata de un campo de trabajo activo. A pesar de ello el marco de estos sistemas no está claramente definido, debido a las diferentes formas de entender lo que es la ayuda informatizada y las áreas de aplicación. [9]

Los diseñadores de *software* y *hardware* están conscientes de lo importante que resulta diseñar máquinas y aplicaciones usables. El crecimiento del número de ordenadores y su establecimiento en todos los ámbitos de la sociedad han influenciado profundos cambios en el *software* en general y en los soportes de ayuda al usuario en particular, estos soportes actualmente constituyen un objetivo crítico en el desarrollo de *software* y *hardware* para el éxito comercial del producto. Se podría pensar que si se hiciera un buen diseño de un sistema interactivo, su uso debería ser completamente intuitivo, sin que el usuario necesite ayuda. Eso sería una situación ideal, que está lejos de la realidad en los sistemas actuales, más si se tiene en cuenta que los nuevos paradigmas de interacción han de definir su propio modo de dar soporte y ayuda al usuario. Cuando se vende un sistema informático se facilita al usuario un manual de uso en formato digital o duro, por tanto, siempre se ha de tener en cuenta en el desarrollo de aplicaciones informáticas el diseño del sistema de ayuda como parte fundamental dentro del sistema.

Para adentrarse en el tema de los sistemas de ayuda para entornos informáticos es primordial hacer referencia al significado de las palabras “sistema” y “ayuda” como términos informáticos. Según el Diccionario de informática e internet de Microsoft, un “sistema” es cualquier colección de elementos

⁴ Sistemas operativos, aplicaciones y dispositivos informáticos.

que trabajan conjuntamente para llevar a cabo una tarea. Ejemplos: un sistema operativo que consta de un grupo de programas y archivos de datos; o un sistema de gestión de bases de datos utilizado para procesar tipos específicos de información [10]. Por su parte una “ayuda” es la capacidad de muchos programas de aplicación de visualizar consejos o instrucciones para el uso de las características cuando así las solicita el usuario, mediante un botón que aparece en pantalla, un elemento de menú o una tecla de función [11]. A partir de este momento los autores se referirán a los Sistemas de Ayuda como SA.

Para Greg Kearsley, los SA para entornos informáticos son programas que asisten al usuario en su utilización, por tanto, su propósito principal es proporcionar apoyo a los problemas específicos que se presentan en el uso de dicha aplicación. Además de esta mejora de rendimiento, deben promover una mayor comprensión del funcionamiento del sistema, que con el tiempo hará disminuir la necesidad de la ayuda. [12]

Teniendo en cuenta el concepto de SA para entornos informáticos de Greg Kearsley, además de los conceptos relacionados con “sistema” y “ayuda”. Los autores de la presente investigación definen que los SA para entornos informáticos son: Las aplicaciones que asisten al usuario como apoyo para mejorar el aprendizaje y rendimiento del trabajo con entornos informáticos.

1.2 Características de los Sistemas de Ayuda para entornos informáticos.

Atendiendo a la importancia de los SA, estos deben tener ciertas características básicas, destacándose las siguientes:

- Facilitar el manejo de la aplicación por parte de los usuarios y brindar la posibilidad de lograr un mejor entendimiento para obtener mejores resultados en el uso de determinada aplicación informática.
- Contener información sobre conceptos y procedimientos para ejecutar tareas.
- Estar orientados al usuario final.

1.3 Clasificación de los Sistemas de Ayuda.

Teniendo en cuenta el campo de acción de la presente investigación se estudiaron cuatro tipos de SA: Sistemas de Ayudas en Línea (SAL), Sistemas de Ayuda a la Decisión (SAD), Sistemas Tutores Inteligentes (STI) y Sistemas de Ayuda Inteligente (SAI). También existen otras ayudas más sencillas que no llegan a ser un sistema, ejemplo de ellas son las ayudas emergentes (*pop-up*) y las ayuda sensible al contexto (*context-sensitive help*).

- **Sistemas de Ayuda en Línea (SAL).**

Los SAL son parte de la documentación externa de un sistema y suelen clasificarse en documentación en línea, ayuda en línea y tutoriales en línea. Sin embargo, esta categorización en la práctica resulta en la mayoría de las ocasiones sutil, ya que en todo momento el manejo de cualquier aplicación debe tener un soporte dinámico y contextual que ayude a su manejo, integrando las técnicas y necesidades del usuario. Las dos líneas principales de distribución de la información en este campo son las ayudas en línea y el material impreso, además existen otros medios tales como vídeos o cintas de audio. [13]

- **Sistemas de Ayuda a la Decisión (SAD).**

En un sentido amplio, se define al SAD como un conjunto de programas y herramientas que permiten obtener oportunamente la información requerida durante el proceso de la toma de decisiones, en un ambiente de incertidumbre. A lo anterior se agrega que, en la mayoría de los casos lo que constituye el detonante de una decisión es el tiempo límite o máximo en el que se debe tomar. [14]

- **Sistemas Tutores Inteligentes (STI).**

La idea de aprovechar las herramientas informáticas en la enseñanza se remonta a los años 50, pero no fue hasta los 80 cuando surgieron los STI con la vocación de desarrollar procesos de enseñanza adaptados a los diferentes usuarios (estudiantes). Son sistemas que contienen una o varias lecciones tutoriales implementadas en una computadora, que al interactuar con el estudiante tienen un comportamiento inteligente. [15]

- **Sistemas de Ayuda Inteligentes (SAI).**

Los SAI surgieron en los años 80 como una propuesta para mejorar el aprendizaje y el rendimiento en la utilización de entornos informáticos. Habitualmente la ayuda tiene como único fin la mejora del rendimiento del usuario, pero dicha asistencia debe tener también un propósito educativo. Si el usuario mejora su conocimiento sobre la aplicación, su necesidad de ayuda decrecerá con el tiempo. El diseño de un SAI es un proceso complejo en el que existen muchas alternativas de diseño que influyen en sus funcionalidades y factibilidad en dominios reales.

Las más importantes son:

- a) ¿Quién tiene la iniciativa en la interacción?, el usuario debe solicitar la ayuda (sistema pasivo) o el SAI puede ofrecer ayuda cuando detecte errores potenciales del usuario (sistema activo).
- b) La integración entre el SAI y la aplicación a la que da soporte, integrado o separado, según el acceso o no del SAI al estado interno de la aplicación.

c) Tipo de información proporcionada y tipos de usuarios, según el propósito principal de la información que se proporciona (instructiva, procedimental, explicativa y especificativa) y si se consideran distintos tipos de usuarios según su experiencia o interés. [16]

Las características de los SAI son las siguientes:

- Incorporan técnicas de Inteligencia Artificial.
- Es alta la capacidad de contextualización de la ayuda proporcionada a las circunstancias concretas en que se solicita. (ejemplo: tarea, usuario)
- La arquitectura es la utilizada en los Sistemas Tutores Inteligentes, pero con ciertas modificaciones.
- Si es un SAI activo puede capturar la interacción del usuario con la aplicación, siendo capaz de responder preguntas y de irrumpir para ayudar al usuario cuando hay problemas o cuando se presenta una oportunidad adecuada para enseñarle alguna nueva funcionalidad del entorno.
- Tiene capacidades de adaptación a las circunstancias concretas en las que se solicita la ayuda.
- Están orientados al usuario final y para aplicaciones de escritorio. [17]

1.4 Ejemplos de Sistemas de Ayuda para entornos informáticos.

Los SA para entornos informáticos surgieron con el objetivo de brindarle al usuario soporte a los problemas que estos puedan presentar con las aplicaciones informáticas. A continuación se mencionan sistemas de ayudas creados por agencias, empresas o universidades extranjeras, en Cuba y en la Universidad de las Ciencias Informáticas, con el objetivo de conocer sus principales características.

En el mundo:

- **Aran**, es un SAI para el SO Unix que se activa mediante una interfaz multimodal, simplifica el acceso, la selección y la comprensión por parte del usuario de la información necesaria para resolver su problema. Integra diferentes tecnologías estándar para proporcionar las funcionalidades de ayuda. Los dos objetivos principales del desarrollo de Aran son demostrar la factibilidad de un SAI para un entorno informático complejo y disponer de un marco de trabajo para la investigación en distintos aspectos de la ayuda interactiva, de la estructuración de información, del diseño de interfaces y del modelado del usuario. El núcleo del sistema está formado por la base de conocimiento⁵, donde se representan, mantienen y organizan distintos tipos de conocimientos e información. [18]

⁵Es un tipo especial de base de datos para la gestión del conocimiento. Provee los medios para la recolección, organización y recuperación computarizada de conocimiento.

- **HealthAgents**, SA basado en código libre que tiene como objetivo principal mejorar la ayuda en la toma de decisiones en el diagnóstico del tratamiento de tumores cerebrales, haciendo uso de la tecnología de agentes. Estos agentes actúan sobre una red distribuida de bases de datos locales que contiene espectros de resonancia magnética, análisis de biochips de ADN⁶ e información clínica de los tumores, para generar unos clasificadores de tumores que puedan incorporarse en un Sistema de Ayuda a la Decisión. [19]

En Cuba:

- Sistema de Información Geográfico para el Análisis y Control de las Ayudas a la Navegación en La Bahía de Nuevitas (**SAYNAV**), tiene como fin gestionar y analizar los datos geoespaciales y sus correspondientes alfanuméricos en tiempo real. Disminuye el tiempo de la toma de decisión y garantiza una visión abarcadora de la situación hidrográfica y de navegación, así como de instalaciones marítimo portuaria en la Bahía de Nuevitas. [20]
- Sistema Informático de Apoyo al Diagnóstico Empresarial (**SIADE**), aplicación que utiliza como base metodológica varias técnicas presentes en la metodología SISTRAT⁷. Permite gestionar y manipular grandes volúmenes de informaciones organizadas en una base de datos, además elaborar consultas, informes, menús y proyectos sobre una o varias tablas. [21]

En la Universidad de las Ciencias Informáticas:

- Sistema de ayuda médico para la Atención de las Dislipoproteinemias (**alasLIPOv2.0**), es un sistema web, de apoyo a los médicos en la toma de decisiones para emitir diagnósticos y tratamientos adecuados a las características de los pacientes afectados por estas enfermedades. [22]
- Ayuda en línea para el sistema **SENTAI** de la Corporación de Importadores y Exportadores de Cuba (CIMEX), brinda información referente al funcionamiento de los módulos que conforman el sistema, así como las relaciones entre estos para facilitar y mejorar el trabajo de los usuarios con el sistema. [23]

1.5 Los Sistemas de Ayuda para sistemas operativos.

Un sistema operativo, “es aquel conjunto coherente de código que gestiona el *hardware* de un sistema de cómputo, creando facilidades para la ejecución en este, de todo tipo de programas” [24]. En el mundo se utilizan disímiles sistemas operativos en dependencia de sus características, ventajas y fin para el que fueron creados, los más utilizados son Windows, Mac, iOS y Linux. La mayoría de los

⁶ Ácido nucleico que contiene la información genética de un ser vivo.

⁷ Sistema de Información de apoyo a la formulación de estrategias.

sistemas operativos poseen su SA para hacerle más eficiente a sus usuarios el uso y desempeño con el mismo. A continuación se describen las principales características, ventajas y funcionalidades de la Ayuda de los SO Fedora, Debian, Ubuntu, Windows XP y Windows 7, por ser los más utilizados en la actualidad. [25]

- **Sistemas operativos libres.**

Las distribuciones Fedora, Ubuntu y Debian utilizan la ayuda de GNOME (*Yelp*), por ello poseen ayudas muy semejantes. Una de las diferencias que tienen es que Ubuntu y Debian la personalizan en un pequeño grado. Las principales características de esta ayuda es que tiene un buscador, botones para ir atrás y adelante, y la información está organizada por temáticas o categorías. La información se presenta de forma estática y en ocasiones mediante listas de temas muy largas. Esta información contiene mucho texto y pocas imágenes que imposibilitan al usuario una mejor recepción de la ayuda brindada.

- **Sistemas operativos propietarios.**

A diferencia de los sistemas operativos libres, los sistemas operativos Windows XP y Windows 7 sí poseen diferencias significativas en su SA, permitiendo delimitar a continuación las principales características.

Características del SA de Windows XP.

- Es nombrado Centro de Ayuda y Soporte.
- La interfaz es atractiva por los colores, imágenes e iconos que utiliza.
- Brinda una ayuda inteligente al usuario, guiándolo por las posibles soluciones.
- Posee un Índice, que brinda la información que se encuentra disponible en la ayuda.
- Cuenta con los botones de adelante y atrás, para ir a las opciones vistas con anterioridad.
- Permite adicionar a “**Favoritos**” los temas vistos, así como ver el “**Historial**” de las acciones realizadas en la ayuda.
- Brinda la posibilidad de utilizar combinaciones de teclas para acceder a las opciones fundamentales.
- Mediante la opción “**Cambiar vista**” permite reducir la interfaz gráfica de la Ayuda a la parte que contiene la información relacionada con el tema en cuestión.
- También permite conocer las características del sistema mediante la opción “**Soporte**”.

Características del SA de Windows 7.

- Es nombrada Ayuda y Soporte técnico de Windows.
- La interfaz no es amena, cuenta con bastante texto y pocas imágenes.
- No posee combinaciones de teclas para acceder a las diferentes opciones.
- Al abrir la interfaz principal necesita conectarse al sitio oficial de Windows en internet para mostrar información actualizada.
- Mediante la opción de **“Explorar la Ayuda”**, se accede a los principales contenidos.
- Algunos de los **“Temas”** poseen la opción de **“Mostrar todo”** u **“Ocultar todo”**, que muestra u oculta toda la información que se encuentre disponible del tema.
- Tiene dos opciones de ayuda, Ayuda sin y con conexión a Internet.

Estos SA poseen características comunes, entre las que se destacan que cuentan con un potente buscador; admiten imprimir contenidos completos, textos seleccionados e imágenes. También permiten obtener más información sobre los diferentes temas mediante la conexión a internet.

1.6 Tipo de Sistema de Ayuda a utilizar.

Los SA se aplican en muy variados campos, proporcionando asistencia a los usuarios en situaciones muy distintas. Por ejemplo, existen sistemas de ayuda tanto para las instalaciones industriales complejas como para la navegación aérea o la toma de decisiones bancarias. Para decidir el tipo de SA a implementar se tuvo en cuenta los cuatro tipos estudiados en el epígrafe 1.3.

De este epígrafe se resume que los Sistemas de Ayuda a la Decisión están enfocados a empresas que necesitan tomar decisiones sobre situaciones relacionadas con la economía. Los Sistemas de Ayudas en Línea son parte de la documentación externa de un sistema, herramienta o aplicación, que puede ser un Manual de uso, Manual de usuario o un Vídeo tutorial de cómo utilizar dicho sistema, herramienta o aplicación. Por su parte los Sistemas Tutores Inteligentes cumplen un objetivo educacional, enfocado en el alumno y no en los usuarios de computadoras. Dada la relación que existe entre el marco de aplicación del presente trabajo y los Sistemas de Ayuda Inteligentes (SAI) en cuanto a sus características y factibilidad de adaptación a los objetivos de esta investigación; los autores deciden que el tipo de SA a implementar para la edición Nova Escritorio será del tipo SAI y para su desarrollo se tendrán en cuenta las características de las ayudas de los sistemas operativos analizados en el epígrafe 1.5.

1.7 Tecnología a utilizar.

El desarrollo de *software* se inició con las aplicaciones de escritorio cuyo rendimiento depende principalmente de las configuraciones y características del *hardware*. Estas aplicaciones habitualmente en su ejecución no requieren comunicación con el exterior, esto repercute en mayor velocidad de procesamiento y tiempo de respuesta, y por tanto en mayores capacidades a la hora de programar herramientas más complicadas. También este tipo de aplicación, permite ser instalada y reinstalada, se adapta a la configuración de la interfaz del sistema operativo y no depende de factores externos, como lo es la red.

Sin embargo, con el advenimiento de la internet y el comercio en línea comenzó el desarrollo por todo el mundo de las aplicaciones web, que se destacan por ser multiplataforma, no requieren actualizaciones, consumen pocos recursos del equipo en el que están instaladas y su funcionalidad es independiente del sistema operativo existente en el ordenador del usuario. En ocasiones son ineficientes debido a que requieren conexión a la red, se corre el riesgo de incompatibilidad con los distintos navegadores y su tiempo de respuesta puede ser muy lento.

Los autores de esta investigación deciden desarrollar el SA para un entorno de escritorio teniendo en cuenta los elementos antes expuestos, además que es la solicitud del cliente y es posible utilizar programas y lenguajes nativos del SO Nova. Otros factores importantes es que dos de los SAI estudiados: Argos y Aran, fueron creados como aplicación de escritorio para el SO Unix.

1.8 Lenguajes utilizados en la implementación de Sistemas de Ayuda.

Para realizar un *software* es necesario realizar un estudio de herramientas, librerías, paradigmas, *framework* y lenguajes de programación. En la actualidad existen varios lenguajes de programación que se utilizan para implementar SA para entornos de escritorio, destacándose C++, Java y *Python*. A continuación se realiza un resumen de las principales características de estos lenguajes.

- **C++**, fue diseñado para extender al exitoso lenguaje de programación C, con mecanismos que permiten la manipulación de objetos. Es un lenguaje híbrido, de propósito general, al que se han añadido nuevos tipos de datos, clases, plantillas y mecanismos de excepciones [26]. Este lenguaje fue utilizado para crear los SA Argos y Aran.
- **Java**, es orientado a objeto, desarrollado con la idea original de ser usado para crear páginas web aunque en la actualidad se utiliza fundamentalmente para desarrollar aplicaciones de escritorio. Con la programación en Java, se pueden realizar *applets*, procesadores de palabras, hojas de cálculos, aplicaciones gráficas, entre otros. [27]

- **Python**, es un lenguaje de scripts interpretado, de alto nivel y de propósito general. Es uno de los lenguajes que más está influyendo en la escena de los programadores. Es de código abierto, multiplataforma y resulta adecuado para manipular archivos de texto. Posee infinidad de librerías que permiten trabajar desde diseño dinámico de sitios web hasta la programación de escritorio o diseño de videojuegos. [28]

Para implementar el SA los autores deciden utilizar **Python** como lenguaje de programación principal, teniendo en cuenta que cumple con las necesidades del presente trabajo, está disponible en los repositorios de Nova, es potente en el manejo de archivos de texto y es el lenguaje propuesto por el cliente. Para complementar las características del SA a implementar se utilizará Prolog, y GTK⁸ para desarrollar las interfaces gráficas de usuario (GUI) del SA, además se empleará HTML (*"Hypertext Markup Language"*; en español "Lenguaje de Marcado de Hipertexto") como lenguaje de hipertexto y CSS (*"Cascading Style Sheets"*; en español "Hojas de Estilo en Cascada") para dar estilos a las páginas web que se mostrarán al usuario.

1.9 Herramientas utilizadas en la implementación de Sistemas de Ayuda.

Para crear aplicaciones informáticas es necesario integrar los resultados de diferentes herramientas de tipo informático. *Microsoft HTML Help Workshop*, *ROBOHTML* y *Help Development* son ejemplos de herramientas para crear ayudas para entornos informáticos web. También existen otros programas que se utilizan para implementar ayudas de escritorio.

Para desarrollar el SA se utilizará el NetBeans IDE 6.7, el motor de inferencia de SWI-Prolog⁹ para inferir a través de preguntas el posible problema que presente el usuario con el SO Nova. Además, para complementar aspectos relacionados con el tratamiento de imágenes, el modelado UML y diseño de la interfaz del SA se empleará el GIMP (*"GNU Image Manipulation Program"*; en español "Programa de Edición de Imágenes"), la herramienta CASE¹⁰ Visual Paradigm y Glade¹¹ respectivamente.

1.10 Aspectos arquitectónicos del Sistema de Ayuda.

Luego de definir el tipo de tecnología, lenguajes y herramientas a emplear en el desarrollo del SA, se hace necesario esclarecer aspectos relacionados con la arquitectura de *software*. Esta es la forma en la que se organizan los componentes de un sistema, interactúan y se relacionan entre sí y con el

⁸ *GIMP Tool Kit*, es una biblioteca del equipo GTK+, que contiene los objetos y funciones para crear la interfaz gráfica de usuario. Maneja widgets como ventanas, botones, menús, etiquetas, deslizadores, pestañas, etc.

⁹ Aplicación de código abierto del lenguaje de programación Prolog. SWI significa "*Sociaal Wetenschappelijke Informatica*"; en español "Informática de Ciencias Sociales").

¹⁰ "*Computer Aided Software Engineering*"; en español "Ingeniería de Software Asistida por Ordenador".

¹¹ Se utiliza para el desarrollo de interfaces de usuario para GTK.

contexto, aplicando normas y principios de diseño y calidad, que fortalezcan y fomenten la usabilidad a la vez que dejan preparado el sistema, para su propia evolución. Dos de las tendencias arquitectónicas utilizadas en la actualidad son: la Arquitectura de Software Orientada a Objetos y la Arquitectura Estructurada, para el diseño del SA se empleará la Arquitectura de Software Orientada a Objetos.

Los tres aspectos arquitectónicos fundamentales en el diseño de la arquitectura de un *software* son: Estilo Arquitectónico, Patrón Arquitectónico y Patrón de Diseño, que representan de lo general a lo particular los niveles de abstracción que componen la Arquitectura del Software.

Según Pressman, los estilos arquitectónicos se enmarcan en cinco categorías generales: Arquitecturas centradas de datos, Arquitecturas de flujo de datos, **Arquitecturas de llamada y retorno**, Arquitecturas orientadas a objetos y Arquitecturas estratificadas [29]. Mientras que en los inicios de la arquitectura de *software* se alentaba la idea de que todas las estructuras posibles en diseño serían susceptibles de reducirse a una media docena de estilos básicos, lo que en realidad sucedió fue que los estilos se tornaron más detallados y exhaustivos. En el trabajo de diploma “Modelo para la ayuda en la toma de decisiones relativas a estilos y patrones arquitectónicos” [30], agrupan en la categoría de Arquitecturas de llamada y retorno a las arquitecturas siguientes: Arquitectura Orientada a Objetos, Arquitectura en Capas (Arquitectura en 3 capas), Arquitectura Modelo-Vista-Controlador (MVC) y Arquitectura basadas en Componentes.

Uno de los estilos arquitectónicos, también conocido como arquitectura y patrón arquitectónico es la Arquitectura en Capas (N-Capas o *N-Layer*). Su ventaja principal es que permite llevar a cabo el desarrollo en varias capas indistintamente. El estilo N-Capas (3 Capas), se divide en las capas de presentación, negocio y datos, como se muestra en la figura 1.1. Estas capas tienen responsabilidades propias que permiten diseñar aplicaciones con un alto nivel de seguridad.

- La capa de presentación (capa de usuario), muestra el sistema al usuario, captura sus solicitudes y presenta los resultados de sus solicitudes. Se comunica con la capa de negocio.
- La capa de negocio (lógica del negocio), recibe las peticiones del usuario, es donde se establecen las reglas del negocio que deben cumplirse. Se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos para solicitar almacenar o recuperar información.
- La capa de datos, es la encargada de acceder (almacenar o recuperar los datos) a la información. Se comunica con la capa de negocio para recibir las solicitudes y brindarle la(s) respuesta(s) a las solicitudes.

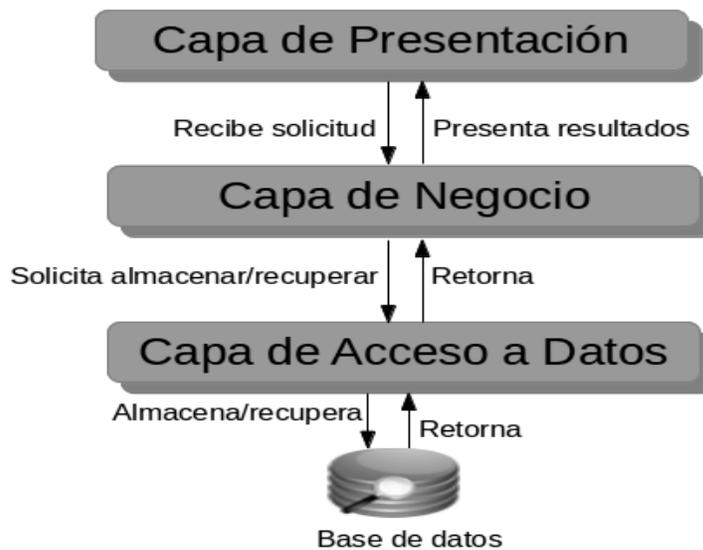


Figura 1.1: Arquitectura *N Layer* (3 Capas).

Para el diseño de la solución que se propone en el trabajo de diploma, se utilizará como arquitectura en los dos subsistemas, la Arquitectura N-Capas a un nivel. El estilo arquitectónico (que pertenece a las Arquitecturas de llamada y retorno) y patrón arquitectónico será *N-Capas*, específicamente se utilizarán 3 Capas. También se aplicarán los patrones de diseño Experto en información y Controlador pertenecientes al patrón **GRASP** (“*General Responsibility Assignment Software Patterns*”; en español “Patrón de *Software* para Asignación de Responsabilidades de uso General”). El estilo de código de programación que se utilizará será PEP-8 (“*Python Enhancement Proposal*”; en español “Propuesta de mejora en Python”) [31].

1.11 Metodología para el desarrollo del Sistema de Ayuda.

La metodología de ingeniería de *software* es un enfoque estructurado, cuya finalidad es hacer más eficaz la producción y lograr alta calidad de una forma costeable. A nivel mundial existen diferentes metodologías para el desarrollo de *software*, como RUP (“*Rational Unified Process*”; en español “Proceso Unificado Racional”) y las Metodologías Ágiles. En el presente trabajo de diploma se utilizará la metodología ágil SXP atendiendo a que es la metodología aplicada en el Departamento SIMAYS¹². SXP es un híbrido de metodologías ágiles que toma las mejores prácticas de las metodologías SCRUM y XP, y se rige por los lineamientos de calidad de la UCI.

SXP consta de cuatro fases principales **Planificación-Definición** donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto; **Desarrollo**, es donde se

¹² Servicios Integrales de Migración, Asesoría y Soporte.

realiza la implementación del sistema hasta que esté listo para ser entregado; **Entrega**, puesta en marcha; y por último **Mantenimiento**, donde se realiza el soporte para el cliente. En estas fases se realizan actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto (LRP), definición de las Historias de Usuario, diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, lo que permite mejorar el diseño cada vez que se le añada una nueva funcionalidad. [32]

1.12 Modelo para desarrollar Sistemas de Ayuda Inteligente.

Para la producción de sistemas de ayuda no existen modelos estándar aplicables a su desarrollo. Por esta razón se realizó un estudio del modelo propuesto en la Tesis Doctoral “Desarrollo de sistemas de ayuda inteligente mediante integración de tecnologías y reutilización de información” [33]. El objetivo fundamental de este modelo es la obtención de sistemas de ayudas eficaces y aplicables a dominios reales. Es un modelo para la construcción de sistemas de ayuda pasivos, que proporcionen información sobre cada aplicación concreta. Las dos ideas en las que se basa la factibilidad del modelo son la integración en un mismo sistema de diferentes tecnologías y la reutilización de información de distintos tipos.

Características del modelo.

- **Sistema activado por el usuario:** En un sistema de ayuda pasivo la interacción tiene que estar promovida por el usuario. Esto podría considerarse una limitación, pues, por ejemplo, podría ocurrir que el usuario no fuera consciente de que tiene problemas o que utilizara métodos muy ineficientes para realizar una determinada tarea. Sin embargo, la alternativa de un sistema de ayuda activo presenta problemas que limitan su aplicación real.
- **Ayuda basada en el acceso a la documentación:** El SA que propone este modelo facilita el acceso a la documentación sobre la aplicación o programa. En lugar de dar información detallada, paso a paso, proporciona información de referencia sobre el programa y su funcionalidad.
- **Adaptación al usuario:** La adaptación del asistente al usuario se realiza por medio de un modelo de usuario basado en las características más significativas de los usuarios.
- **Interfaz multimodal:** En los SA, la interfaz juega un papel fundamental en la aceptación por parte de los usuarios y en la facilidad de uso. Independientemente de la sofisticación de las técnicas incluidas, no es realista considerar que pueda solucionar todas las solicitudes de los usuarios y sin embargo seguir siendo útil.

Integración de tecnologías en el modelo.

En el modelo que se está haciendo mención se ofrecen al usuario funcionalidades de ayuda mediante la combinación de diversas técnicas en un mismo sistema. Estas tecnologías son: recuperación de información, hipertexto, modelado de usuario y representación explícita del conocimiento, como se muestra en la figura 1.2.

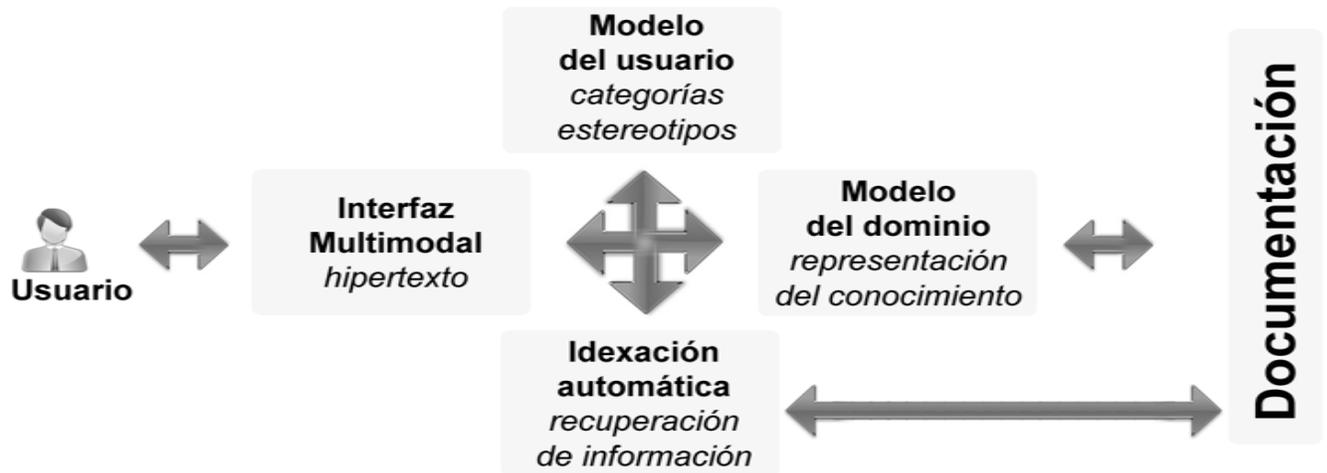


Figura 1.2: Esquema del modelo utilizado en los Sistemas de Ayuda Inteligentes, Argos y Aran.

- **Interfaz Multimodal** (hipertexto): En el diseño de la interfaz es preciso tener presentes los criterios de una adecuada interacción hombre-máquina, de forma que se obtenga una interfaz gráfica que además de ser versátil y sencilla de utilizar, permita distintos modos de operación. En la interacción con los usuarios en el modelo se aplican técnicas de hipertexto debido a dos razones: su facilidad de uso y su adecuación para presentar información sobre dominios amplios y diversos.
- **Modelado de usuario** (categorías y estereotipos): Un Sistema de Ayuda Inteligente debe presentar, en general, un comportamiento adaptativo, adecuando la información que proporcione al interés y el conocimiento del usuario. Para lograr este objetivo de interacción “inteligente”, de modo que el sistema sea capaz de exhibir un comportamiento cooperativo de amplio rango.
- **Modelo del dominio** (representación del conocimiento): En los asistentes inteligentes es habitual encontrar una representación explícita del conocimiento que proporcione un soporte para otros módulos del sistema, facilitando la generación de respuestas mediante la Inteligencia Artificial en un lenguaje natural.
- **Indexación automática** (recuperación de información): Permite recuperar documentos a partir de una consulta. Esta técnica hace posible que el usuario exprese su necesidad con un lenguaje natural, especificando la información que desea obtener sin tener que aprender algún otro método formal de petición.[34]

Capítulo 2: “Sistema de Ayuda para la edición Nova Escritorio”.

En el capítulo anterior se realizó el estudio del arte de los sistemas de ayuda para sistemas operativos, sus características, modelo, lenguajes y herramientas para su desarrollo, así como los diferentes tipos. En el capítulo 2 se hace alusión a la solución propuesta, planteándose cómo se va a desarrollar la misma y se definen las funcionalidades a implementar. Además, se describe el proceso ágil basado en las historias de usuario.

2.1 Sistema de Ayuda propuesto.

Desarrollar un SA es una tarea compleja, teniendo en cuenta el tipo de sistema, la tecnología a utilizar y los fines para los que son creados. En la propuesta de solución al problema relacionado con la carencia de un SA para la edición Nova Escritorio, se propone desarrollar un SA del tipo Sistema de Ayuda Inteligente (pasivo), que utilice características de los Sistemas Basados en el Conocimiento (SBC); teniendo en cuenta los fundamentos y características que se encuentran en los epígrafes 1.3 y 1.6. La propuesta está compuesta por dos subsistemas, Sistema de Gestión (Gestionar información del Sistema de ayuda) y Sistema cliente (Ayuda y Soporte) como se muestra en la figura 2.1.

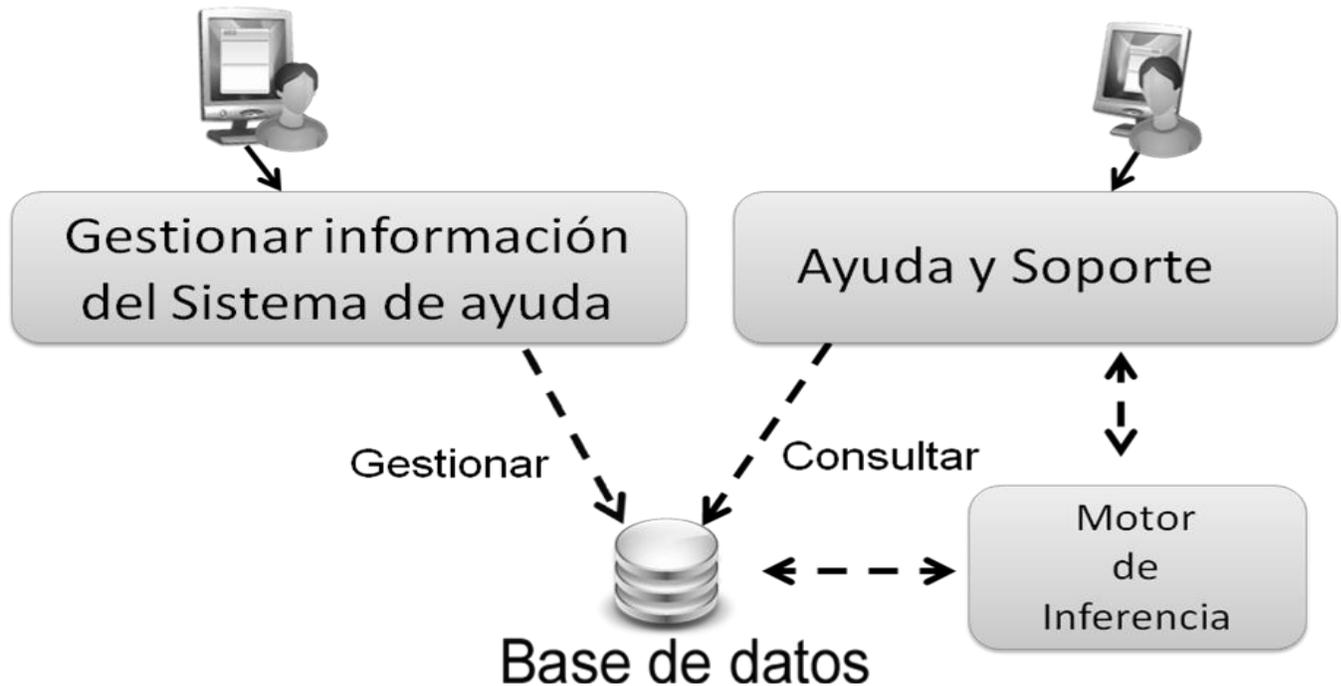


Figura 2.1: Representación del funcionamiento del Sistema de Ayuda.

El subsistema Sistema de gestión (SG) está concebido para gestionar toda la información que contendrá el subsistema Sistema cliente (SC). La información se divide en categorías, subcategorías, problemas y su solución, preguntas frecuentes, reglas y trucos, preguntas y consejos; a continuación se describe cómo serán utilizados.

Las categorías representan los temas generales que se abordan en la ayuda (Multimedia, Red), las subcategorías están relacionadas con la categoría (Audio pertenece a Multimedia, Red cableada a Red), los problemas y su solución contienen los pasos para solucionar el problema. También se utilizarán preguntas frecuentes, que por su importancia y a consideración de los expertos se ubicarán en una sección diferente al de las preguntas normales, estas últimas permitirán identificar los problemas que pueda tener el usuario y se enmarcan en una subcategoría específica.

Para que el SC sea capaz de inferir el problema correcto a mostrar al usuario, teniendo en cuenta sus elecciones tanto de la categoría, subcategoría y de las preguntas, se hace necesario gestionar las reglas¹³ que serán utilizadas por el motor de inferencia para obtener la solución que se brindará al usuario, estas están conformadas por el nombre de la regla (nombre del problema), categoría, subcategoría y pregunta(s), ejemplo de una regla es la siguiente: **no_hay_conexion:-redes, redes_cableadas,como_configurar_la_red.**, cuyo nombre del problema es “No hay conexión”, la categoría es “Redes”, la subcategoría es “Redes cableadas” y la pregunta que identificará el problema es “¿Cómo configurar la red?”.

También se gestionarán trucos y consejos relacionados con el proyecto productivo Nova, combinaciones de teclas y trucos del sistema operativo. Toda esta información formará parte de la base de datos que generará el SG, la base de datos podrá ser modificada por el propio subsistema con el objetivo de lograr mejoras en las soluciones. El SC, consultará la base de datos para mostrar al usuario las categorías, trucos y consejos, y las preguntas frecuentes.

El SA a desarrollar será un sistema de escritorio, basado en el conocimiento y en reglas, del tipo SAI pasivo, que utilizará ficheros para almacenar la información. Para la implementación se utilizarán los lenguajes: Python 2.7, Prolog, HTML 4.0 y CSS 2.1. Como herramientas: Glade 3.10, NetBeans IDE 6.7, Visual Paradigm 8.0, SWI-Prolog 5.0 y GIMP 2.6. Además se emplearán las librerías GTK 3.0, PySWIP en su versión 0.2.2 para lograr un puente de comunicación entre Python y SWI-Prolog, y *unittest* para realizar las pruebas Unitarias. Como arquitectura de *software* se aplicará la arquitectura N-Capas a un nivel, como estilo arquitectónico y patrón arquitectónico N-Capas específicamente se utilizarán 3-Capas y como patrones de diseño, el Experto en información y Controlador pertenecientes al patrón GRASP (“*General Responsibility Assignment Software Patterns*”; en español “Patrón de *Software* para Asignación de Responsabilidades de uso General”). Para definir la propuesta de solución se tuvo en cuenta fundamentalmente los criterios del cliente, los que se obtuvieron a través de las entrevistas que se encuentran en el **Anexo 1**, donde se muestran las principales preguntas realizadas al cliente.

¹³Las reglas se utilizan en Prolog para significar que un hecho depende de uno o más hechos.

2.2 Modelo que se utiliza para desarrollar el Sistema de Ayuda.

En el epígrafe 1.12 se describen las diferentes tecnologías utilizadas en el modelo empleado para crear Sistemas de Ayuda Inteligentes. Para el desarrollo del SA se aplicarán estas tecnologías, pero ajustadas a las necesidades y objetivos de la investigación. Las adaptaciones y cómo se aplicarán estas tecnologías (técnicas) se describen a continuación:

- **Interfaz multimodal** (hipertexto): Los elementos más característicos de un hipertexto son los nodos y los enlaces. El nodo es la unidad de información, en el SA se utilizarán para representar los ficheros que contendrán las reglas, descripción, trucos y consejos, preguntas y problemas con su solución. Los enlaces definen las relaciones establecidas entre los nodos, que se verán reflejados a través de las reglas que representan la conexión entre los nodos.
- **Modelo de usuario** (categorías y estereotipos): En el SA se cumple que el sistema comparte las responsabilidades con el usuario para asegurar el éxito de la comunicación usuario-sistema.
- **Modelo del dominio** (representación del conocimiento): Se aplicará para mostrar los conocimientos (información) al usuario a través de las reglas en el lenguaje de programación Prolog.
- **Indexación automática** (recuperación de información): Se utilizará mediante la funcionalidad de buscar, la cual mostrará el nombre de los problemas o preguntas frecuentes relacionadas con el texto introducido y al conformar el camino de la regla según la selección el usuario.

2.3 Proceso de desarrollo del proyecto basado en SXP.

Con el objetivo de llevar a cabo el proceso de desarrollo de forma organizada, se toma como base la metodología ágil SXP. La cual define varios artefactos, ejemplo de ellos son la tabla de roles, historias de usuario (HU), plan de release y lista de reserva del producto (LRP). La tabla de roles permite el desglose de las responsabilidades de los roles que intervienen en el desarrollo del producto. A continuación se especifican en la tabla 2.1.

Rol	Responsabilidad	Nombre
Gerente	Dirige y controla las tareas del equipo. Toma las decisiones finales, participa en la selección de objetivos y requerimientos. Controla el progreso y da seguimiento a cada iteración. Evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes.	Ing. Yaima Oval Riverón Ing. Susana Sánchez Ortiz Ing. Alfredo Pérez Bénitez

Rol	Responsabilidad	Nombre
Cliente	Participa en las tareas que involucran la lista de reserva del producto y realiza las pruebas de aceptación.	Ing. Daniel Hernández Bahr (Ver Anexo 2)
Programadores	Elabora el código de las funcionalidades a implementar.	José Gustavo Suárez Matilla Yoency Leyva Ochoa
Analista	Es el encargado de escribir las historias de usuario, las pruebas de aceptación y revisar el contenido que se mostrará al usuario.	Yoency Leyva Ochoa José Gustavo Suárez Matilla
Diseñadores	Encargado del diseño del sistema, de los prototipos de interfaces, máximo responsable de la realización del diseño de las metáforas y supervisan el proceso de construcción.	Yoency Leyva Ochoa José Gustavo Suárez Matilla
Encargado de pruebas	Ayuda al cliente a escribir las pruebas de aceptación. Ejecuta las pruebas regularmente y difunde los resultados en el equipo.	Yoency Leyva Ochoa José Gustavo Suárez Matilla
Arquitecto	Se vincula directamente con el analista y el diseñador debido a que su trabajo tiene que ver con la estructura y el diseño del sistema.	Yoency Leyva Ochoa José Gustavo Suárez Matilla

Tabla 2.1: Planificación del proyecto por roles.

2.4 Modelo de dominio.

Un sistema por pequeño que parezca no lo hace sencillo, por ello se establece una técnica para la especificación de los requisitos más importantes del sistema, que va a dar soporte al negocio, “el modelo del negocio”. En dependencia de la situación o escenario se determina si es necesario un modelo completo del negocio o de lo contrario se procede a definir el modelo conceptual o modelo de dominio. El modelo de dominio del presente trabajo se muestra en la figura 2.2.

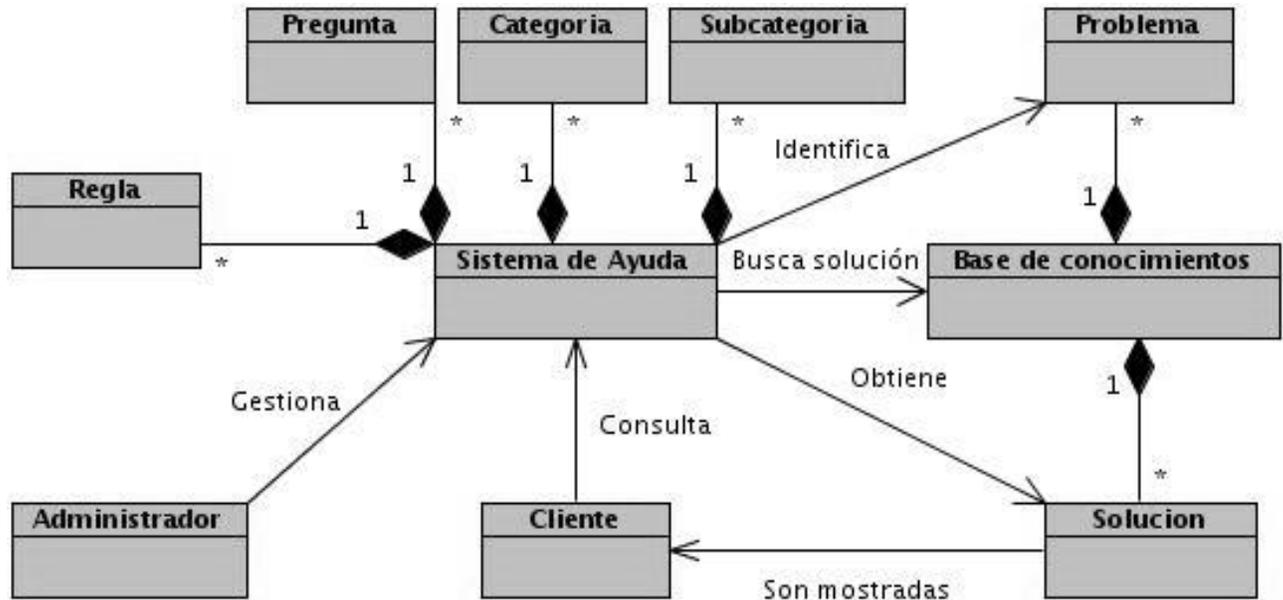


Figura 2.2: Modelo de dominio del Sistema de Ayuda para la edición Nova Escritorio.

2.5 Lista de Reserva del Producto.

La Lista de Reserva del Producto (LRP) está conformada por una lista priorizada que define el trabajo a realizar en el proyecto. Esta lista puede crecer y editarse a medida que se obtienen más conocimientos acerca del producto, con la restricción de que sólo puede cambiarse entre iteraciones. La LRP se muestra en la siguiente tabla.

Prioridad	Ítem *	Descripción	Estimación	Estimado por
Muy Alta				
	1	Adicionar categorías.	2 días	ANA ¹⁴
	2	Mostrar categorías.	2 días	
	3	Editar categorías.	2 días	
	4	Eliminar categorías.	2 días	
	5	Adicionar subcategorías.	2 días	
	6	Mostrar subcategorías.	2 días	
	7	Editar subcategorías.	2 días	
	8	Eliminar subcategorías.	2 días	
	9	Adicionar contenido problemas con su solución o preguntas frecuentes.	4 días	
	10	Mostrar contenido problemas con su solución o preguntas frecuentes.	5 días	
	11	Editar contenido problemas con su solución o preguntas frecuentes.	4 días	
	12	Eliminar contenido problemas con su solución	2 días	

¹⁴Analista.

Prioridad	Ítem *	Descripción	Estimación	Estimado por
		o preguntas frecuentes.		
	13	Adicionar preguntas.	4 días	
	14	Mostrar preguntas.	3 días	
	15	Editar preguntas.	4 días	
	16	Eliminar preguntas.	4 días	
	17	Adicionar reglas.	4 días	
	18	Mostrar reglas.	3 días	
	19	Editar reglas.	4 días	
	20	Eliminar reglas.	4 días	
	21	Adicionar trucos y consejos.	4 días	
	22	Mostrar trucos y consejos.	4 días	
	23	Editar trucos y consejos.	4 días	
	24	Eliminar trucos y consejos.	3 días	
Alta				
	25	Identificar el problema mediante la selección del usuario.	2 días	ANA
	26	Buscar información.	8 días	
	27	Exportar la solución del problema y el contenido de las preguntas frecuentes a formato pdf.	2 días	
	28	Imprimir la solución del problema y el contenido de las preguntas frecuentes.	2 días	
Media				
	29	Mostrar la solución del problema.	4 días	ANA
	30	Mostrar el contenido de las preguntas frecuentes.	5 días	
	31	Mostrar categorías.	7 días	
	32	Mostrar subcategorías.	4 días	
	33	Mostrar preguntas.	5 días	
Baja				
	34	Seleccionar preguntas.	9 días	ANA
	35	Aumentar el tamaño del texto de la solución del problema y del contenido de las preguntas frecuentes.	8 días	
	36	Disminuir el tamaño del texto de la solución del problema y del contenido de las preguntas frecuentes.	3 días	
	37	Restablecer el tamaño del texto de la solución del problema y del contenido de las preguntas frecuentes.	2 días	
	38	Adicionar a favoritos.	8 días	
	39	Eliminar de favoritos.	6 días	

Prioridad	Ítem *	Descripción	Estimación	Estimado por
RNF (Requisitos No Funcionales)				
Usabilidad				
	40	El sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada.		
	41	El sistema debe poseer una interfaz de manejo cómoda, que posibilite a los usuarios sin experiencia una rápida adaptación.		
Restricciones de diseño				
	42	Lenguaje de programación: Python 2.7 y Prolog 5.0.		
	43	Como IDE de programación se empleará NetBeans 6.7.		
	44	El sistema operativo a utilizar en el entorno de desarrollo debe ser Nova.		
	45	SWI-Prolog 5.0 como motor de inferencia.		
Interfaz				
	46	Interfaz amigable, fácil de usar, sencilla e interactiva.		
	47	El diseño gráfico será acorde con las Guías de la interfaz humana de GNOME 2.2.2.		

Tabla 2.2: Lista de Reserva del Producto (LRP).

2.6 Historias de Usuario.

Las historias de usuario son la técnica utilizada en la metodología SXP para especificar los requisitos del *software*, lo que equivaldría a los casos de uso en el proceso unificado. Las historias de usuario (HU) describen las tareas que el sistema debe hacer, cuestión que depende en gran medida de las especificaciones realizadas por el cliente. Van a ser la guía para la construcción posterior de las pruebas de aceptación comprobando de esta manera la correcta implementación de las historias de usuario. En el **Anexo 3** se encuentran las principales HU con sus tareas de ingenierías correspondientes.

2.7 Plan de release.

El plan de release que se encuentra en la tabla 2.3, es una plantilla en la que el cliente decide qué historias de usuario (HU) deben ser incluidas en cada iteración (lanzamiento). Este plan ayuda a definir cuáles son las HU más significativas y permite dividir el proceso de desarrollo de *software* en iteraciones, planificando el trabajo a realizar en cada una de las iteraciones.

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
Iteración 2	En esta iteración se desarrollarán la mayoría de las historias de usuario que tienen prioridad muy alta.	HU_1, HU_2, HU_3	4 semanas
Iteración 3	Se continuará el desarrollo de las historias de usuario de muy alta prioridad y se integrarán con las anteriores para conformar el subsistema de gestión.	HU_4, HU_5, HU_6	4 semanas
Iteración 4	Se desarrollarán las historias de usuario de prioridad alta y algunas de las de prioridad media.	HU_7, HU_8, HU_9, HU_10, HU_11, HU_12, HU_13	4 semanas
Iteración 5	En esta iteración se continuará con el desarrollo de las historias de usuario de prioridad media y baja.	HU_14, HU_15, HU_16, HU_17	4 semanas
Iteración 6	En esta iteración se continuará con el desarrollo de las historias de usuario de prioridad baja y se integrarán con las HU de las iteraciones 4 y 5 para conformar el subsistema cliente.	HU_18, HU_19, HU_20, HU_21	4 semanas

Tabla 2.3: Plan de release.

2.8 Diseño con metáforas.

El Diseño con metáforas es sencillamente el diseño de la solución más simple que pueda funcionar y ser implementado en un momento dado del proyecto. Las metáforas conforman el vocabulario para realizar la descripción del problema, conformando el diseño de solución para determinados momentos del proyecto, generando el Modelo de Diseño, el que está integrado por un diagrama de Paquetes que expone dicho diseño, este diagrama muestra los elementos físicos del sistema y sus relaciones. A continuación se representa en figura 2.3 el diagrama de Paquetes del SA.

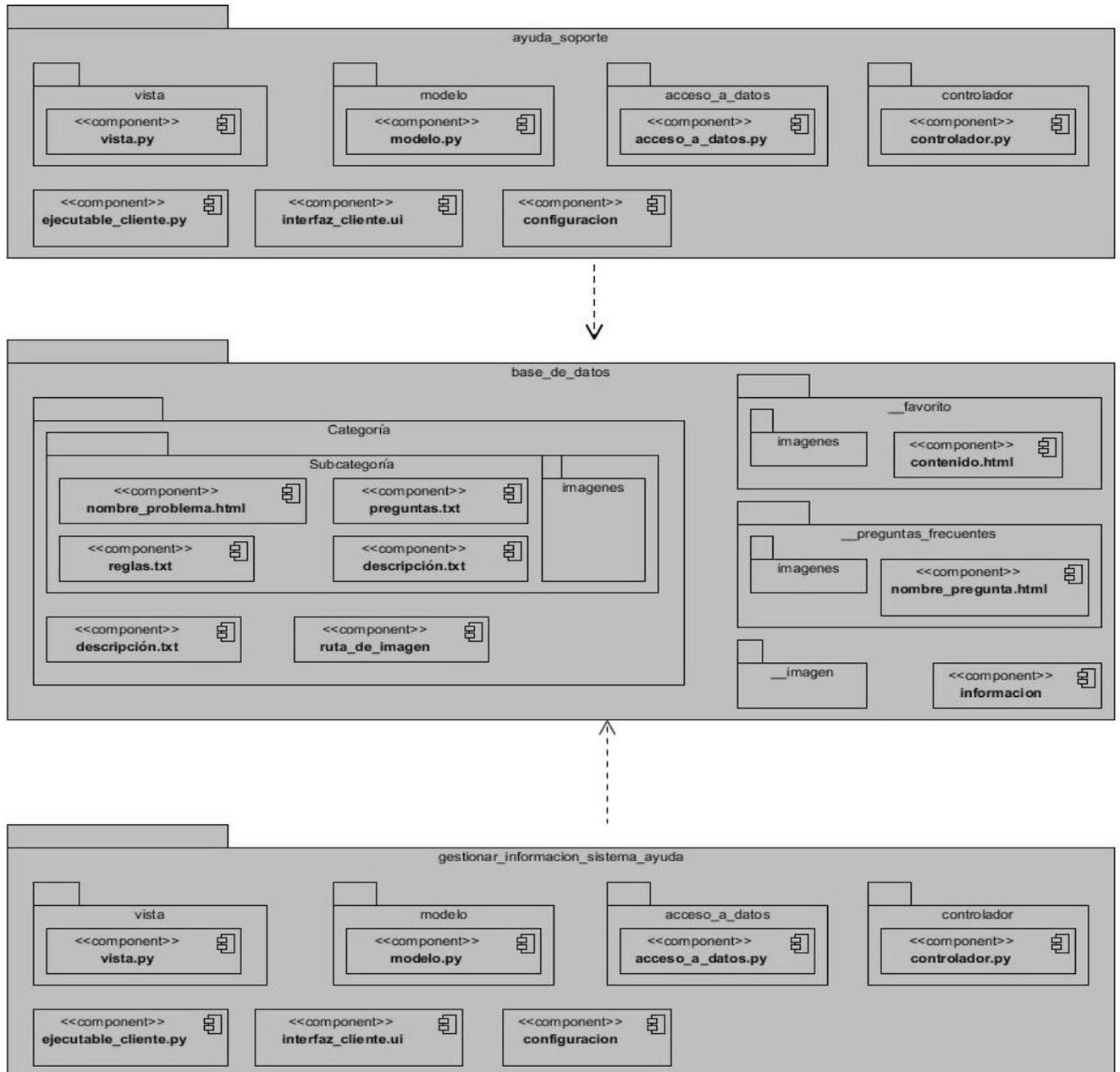


Figura 2.3: Diagrama de Paquetes del Sistema de Ayuda.

Descripción:

El diagrama está compuesto por dos sistemas y una base de datos, el primero “Ayuda y Soporte” y el otro “Gestionar Información del Sistema de Ayuda”. El sistema “Gestionar Información del Sistema de Ayuda” está compuesto por los ficheros **ejecutable_gestion.py** (se emplea para ejecutar el módulo “Gestionar Información del Sistema de Ayuda”), **interfaz_gestion.ui** (contiene los elementos visuales del módulo) y un fichero configuración donde se almacena la dirección donde el sistema crea o carga la base de datos. Además cuenta con los componentes **vista** (contiene el fichero `vista.py`, que posee las clases que se utilizan para la interfaz del usuario), **modelo** (en el fichero `modelo.py` se encuentran las

clases que representan la lógica del negocio), **controlador** (compuesta por el fichero controlador.py que representa las clases controladoras) y **acceso a datos** (compuesta por el fichero acceso_a_datos.py que representan las clases para acceder a los datos).

El sistema de “Ayuda y Soporte” está compuesto por los ficheros **ejecutable_cliente.py** (se emplea para ejecutar el modulo del sistema de “Ayuda y Soporte”), **interfaz_cliente.ui** (contiene los elementos visuales del módulo) y el fichero configuración donde se almacena la dirección donde el sistema crea o carga la base de datos. Además cuenta con los componentes **vista** (contiene el fichero vista.py, que contiene las clases que se utilizan para la interfaz del usuario), **modelo** (en el fichero modelo.py se encuentran las clases que representan la lógica del negocio), **controlador** (compuesta por el fichero controlador.py que representa las clases controladoras) y **acceso a datos** (compuesta por el fichero acceso_a_datos.py que representan las clases para acceder a los datos).

En la base de datos, se representan las categorías, su descripción e imagen, dichas categorías están compuestas por los ficheros **descripción.txt** (describe la categoría), **ruta_de_imagen**(almacena la ruta de la imagen) y subcategorías que contienen los ficheros **nombre_problema.html** (el problema con la solución que se mostrará al usuario), **preguntas.txt** (preguntas que identifican el problema con su solución), **descripción.txt** (describe la subcategoría), **reglas.txt** (contiene las reglas conformadas por nombre, categoría, subcategorías y un conjunto de pregunta(s)) y un directorio utilizado para almacenar las imágenes que son utilizadas en la solución del problema. Existen los contenidos añadidos por el usuario a favoritos que está compuesto por **contenido.html** y un directorio donde se almacenan las imágenes del contenido, preguntas frecuentes que está compuesto por **nombre_pregunta.html** y un directorio donde se almacenan las imágenes de las preguntas y los trucos y consejos que se almacenan en el fichero **información.txt**.

2.9 Diagrama de componentes.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos de *software* que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, entre otros. A continuación se presenta el diagrama de Componentes del SA.

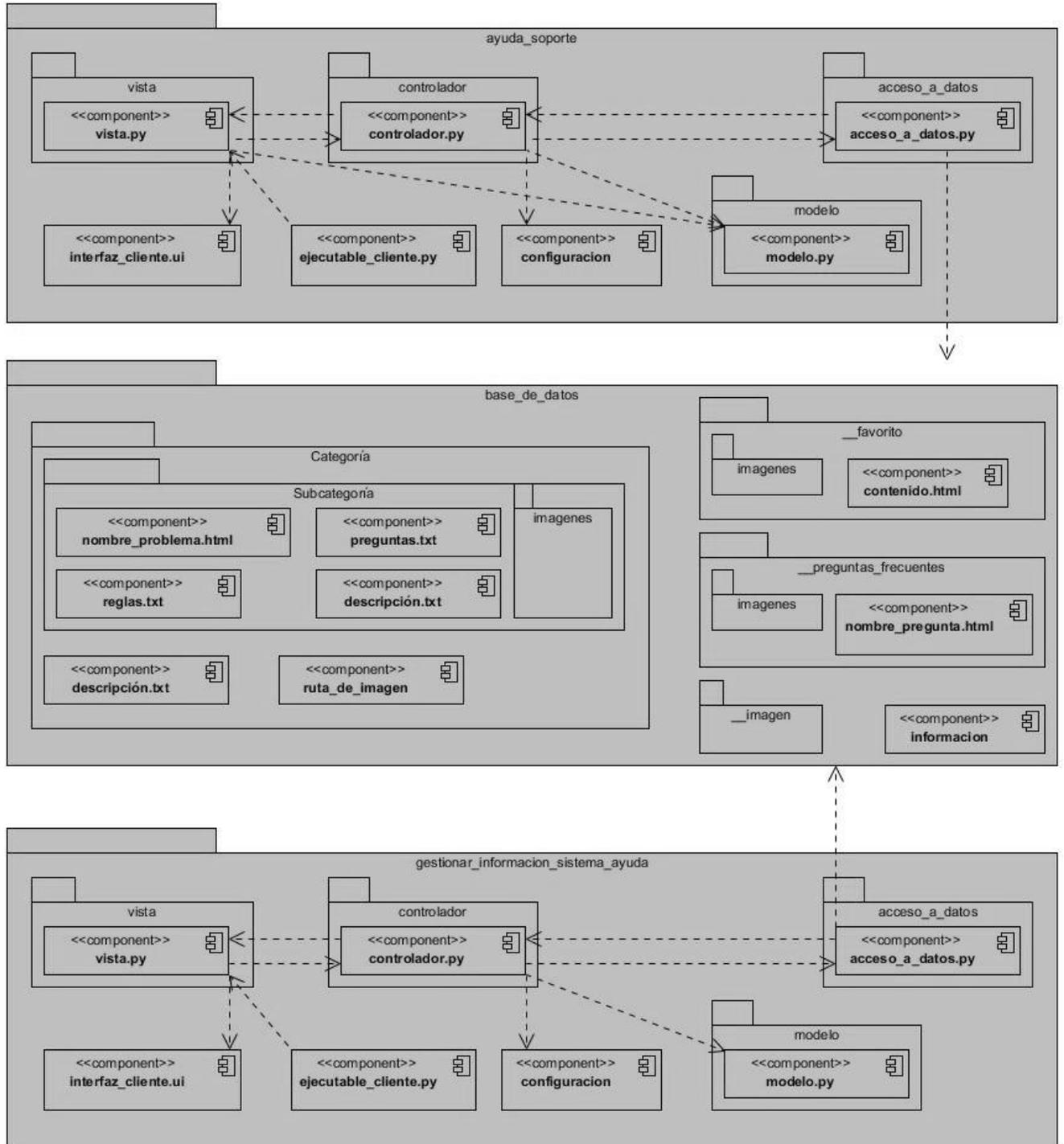


Figura 2.4: Diagrama de Componentes del Sistema de Ayuda.

Capítulo 3: “Diseño y realización de pruebas”.

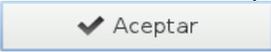
Para asegurar el correcto funcionamiento y calidad del *software* se aplican pruebas de *software*, que son procesos que permiten verificar, revelar la calidad de un producto y son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad. Las pruebas de *software* se dividen según el método, en pruebas de “Caja blanca” que se centran en los detalles procedimentales y en “Caja negra” que se aplican a la interfaz.

Al producto desarrollado se le aplicaron pruebas de **Aceptación** (caja negra) debido a que es el tipo de prueba definido en la metodología SXP. También se realizaron pruebas **Unitarias** (caja blanca) durante el ciclo de desarrollo, utilizando la librería de Python *unittest*, para probar el correcto funcionamiento del sistema.

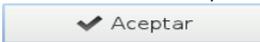
3.1 Pruebas de Aceptación.

Las pruebas de aceptación son definidas por el cliente y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponde al cliente. La utilización de estas permite a los programadores medir la calidad de su trabajo y garantizar la entrega de un producto con calidad y en correspondencia con las necesidades del cliente. Se definieron los casos de prueba de las historias de usuario del SA para la edición Nova Escritorio (SAENE). A continuación se muestran 8 de los 39 casos de prueba que tiene el presente trabajo de diploma.

I. Casos de pruebas para la HU Gestionar categoría.

Caso de Prueba de Aceptación	
Código Caso de Prueba: SAENE-01-1	Nombre Historia de Usuario: Gestionar categoría.
Nombre de la persona que realiza la prueba: Yoency Leyva Ochoa	
Descripción de la Prueba: Consiste en adicionar categorías a la base de datos.	
Condiciones de Ejecución: El usuario tiene que introducir la categoría y su descripción, la elección de una imagen es opcional.	
Entrada / Pasos de ejecución:	
Pasos:	
<ol style="list-style-type: none"> 1. Introducir el nombre de la categoría. 2. Es opcional seleccionar una imagen. 3. Llenar el campo asociado con la descripción de la categoría. 4. Dar clic en el botón . 	
Resultado Esperado: El sistema muestra un mensaje indicando que la categoría ha sido adicionada correctamente.	
Evaluación de la Prueba: Satisfactoria.	

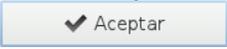
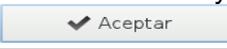
Caso de Prueba de Aceptación	
Código Caso de Prueba: SAENE-01-2	Nombre Historia de Usuario: Gestionar categoría.
Nombre de la persona que realiza la prueba: Yoency Leyva Ochoa	
Descripción de la Prueba: Esta prueba consiste en mostrar las categorías de la base de datos.	
Condiciones de Ejecución: Deben existir categorías en la base de datos.	
Entrada / Pasos de ejecución:	
Pasos:	
1. Dar clic en el botón  de la barra de herramientas.	
Resultado Esperado: El sistema muestra las categorías que tiene almacenada en la base de datos.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: SAENE-01-3	Nombre Historia de Usuario: Gestionar Categoría.
Nombre de la persona que realiza la prueba: Yoency Leyva Ochoa	
Descripción de la Prueba: Esta prueba consiste en editar categorías de la base de datos.	
Condiciones de Ejecución: Deben existir categorías en la base de datos.	
Entrada / Pasos de ejecución:	
Pasos:	
1. El usuario debe seleccionar la categoría que desee editar.	
2. El sistema muestra el nombre de la categoría, la imagen que tenga asociada y la descripción de la misma.	
3. El sistema permite editar el nombre, la imagen y la descripción.	
4. Dar clic en el botón  .	
Resultado Esperado: El sistema muestra un mensaje indicando que la categoría ha sido editada correctamente.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: SAENE-01-4	Nombre Historia de Usuario: Gestionar categoría.
Nombre de la persona que realiza la prueba: Yoency Leyva Ochoa	
Descripción de la Prueba: Esta prueba consiste en eliminar categorías de la base de datos.	
Condiciones de Ejecución: Debe existir al menos una categoría en la base de datos.	
Entrada / Pasos de ejecución:	
Pasos:	
1. Dar clic en el botón  .	
2. El sistema muestra un mensaje preventivo ¿Está seguro que desea eliminar la selección?	
Resultado Esperado: Si el usuario escoge que Si, el sistema muestra un mensaje indicando que la categoría ha sido eliminada correctamente.	
Evaluación de la Prueba: Satisfactoria.	

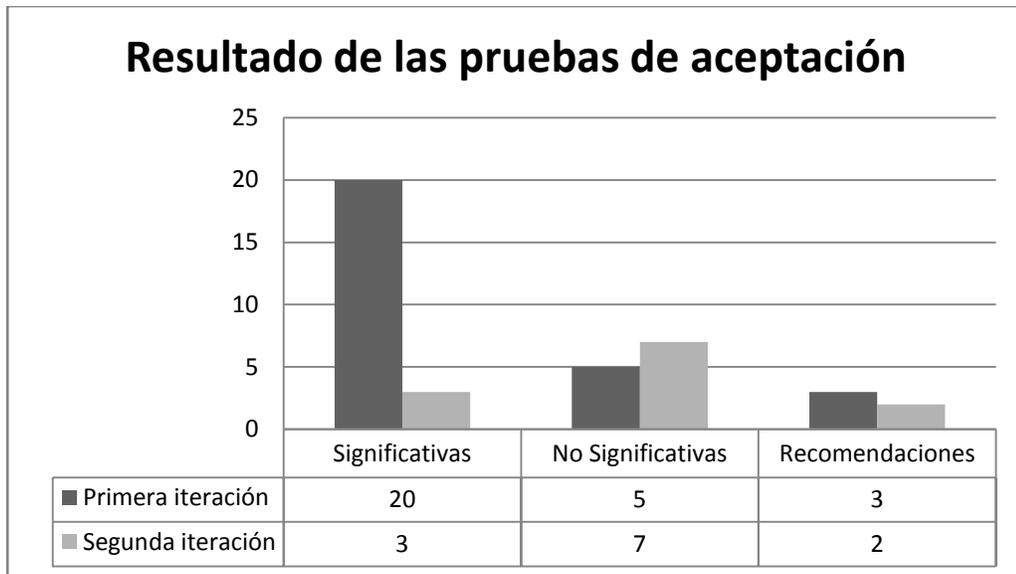
II. Casos de pruebas para la HU Gestionar subcategorías.

Caso de Prueba de Aceptación	
Código Caso de Prueba: SAENE-02-1	Nombre Historia de Usuario: Gestionar subcategorías.
Nombre de la persona que realiza la prueba: Yoency Leyva Ochoa	
Descripción de la Prueba: Esta prueba consiste en adicionar subcategorías a la base de datos.	
Condiciones de Ejecución: Tiene que existir al menos una categoría en la base de datos.	
Entrada / Pasos de ejecución:	
Pasos:	

<ol style="list-style-type: none"> 1. Seleccionar la categoría a la que quiere adicionarle una nueva subcategoría. 2. Escribir el nombre de la subcategoría en el campo “Nombre”. 3. Introducir una pequeña descripción de la categoría en “Descripción”. 5. Dar clic en el botón .
Resultado Esperado: El sistema muestra un mensaje indicando que la subcategoría ha sido adicionada correctamente.
Evaluación de la Prueba: Satisfactoria.
Caso de Prueba de Aceptación
Código Caso de Prueba: SAENE-02-2 Nombre Historia de Usuario: Gestionar subcategorías.
Nombre de la persona que realiza la prueba: Yoency Leyva Ochoa
Descripción de la Prueba: Esta prueba consiste en mostrar subcategorías de la base de datos.
Condiciones de Ejecución: Deben existir categorías y subcategorías en la base de datos.
Entrada / Pasos de ejecución
Pasos:
<ol style="list-style-type: none"> 1. Seleccionar la categoría de la cual quiere mostrar su(s) subcategoría(s).
Resultado Esperado: El sistema muestra las subcategorías con su descripción, pertenecientes a la categoría seleccionada por el usuario.
Evaluación de la Prueba: Satisfactoria
Caso de Prueba de Aceptación
Código Caso de Prueba: SAENE-02-3 Nombre Historia de Usuario: Gestionar subcategorías.
Nombre de la persona que realiza la prueba: Yoency Leyva Ochoa
Descripción de la Prueba: Esta prueba consiste en editar subcategorías de la base de datos.
Condiciones de Ejecución: Deben existir categorías y subcategorías en la base de datos.
Entrada / Pasos de ejecución:
Pasos:
<ol style="list-style-type: none"> 1. Seleccionar la categoría a que pertenece la subcategoría se desea editar. 2. Seleccionar la subcategoría. 3. El sistema permite editar el nombre y la descripción. 4. Dar clic en el botón .
Resultado Esperado: El sistema muestra un mensaje indicando que la subcategoría ha sido editada correctamente.
Evaluación de la Prueba: Satisfactoria.
Caso de Prueba de Aceptación
Código Caso de Prueba: SAENE-02-4 Nombre Historia de Usuario: Gestionar Subcategorías.
Nombre de la persona que realiza la prueba: Yoency Leyva Ochoa
Descripción de la Prueba: Esta prueba consiste en eliminar subcategorías de la base de datos.
Condiciones de Ejecución: Deben existir categorías y subcategorías en la base de datos.
Entrada / Pasos de ejecución: Pasos:
<ol style="list-style-type: none"> 1. Seleccionar la categoría a la que pertenece la subcategoría que se desea eliminar. 2. Seleccionar la subcategoría a eliminar. 3. Dar clic en el botón . 4. El sistema muestra un mensaje preventivo ¿Está seguro que desea eliminar la selección?
Resultado Esperado: Si el usuario escoge que Si, el sistema muestra un mensaje indicando que la subcategoría ha sido eliminada correctamente.
Evaluación de la Prueba: Satisfactoria.

3.2 Resultados de las pruebas de Aceptación.

Las pruebas de Aceptación se realizaron en dos iteraciones, en la primera se obtuvo como resultado un total de 28 No Conformidades (NC), de ellas 20 significativas, 5 no significativas y 3 recomendaciones. Luego se efectuó una segunda iteración de pruebas, que arrojó como resultado, 12 No conformidades, de ellas 3 significativas, 7 no significativas y 2 recomendaciones. Todas estas NC quedaron resueltas por parte del equipo de desarrollo, estos resultados se representan mediante la siguiente gráfica.



Para certificar la calidad del SA para la edición Nova Escritorio, el jefe del Departamento SIMAYS y el cliente principal certificaron a través de las cartas de aceptación su satisfacción con el producto. Las copias de estas cartas se encuentran en el **Anexo 4** y **Anexo 5**.

3.3 Verificación del Sistema de Ayuda Inteligente.

La verificación de un sistema inteligente es un proceso que incluye las siguientes tareas:

- Verificación del cumplimiento de las especificaciones.
- Verificación de los mecanismos de inferencia.
- Verificación de la base de conocimientos.

Verificación del cumplimiento de las especificaciones.

El análisis del cumplimiento de las especificaciones puede ser llevado a cabo por los desarrolladores, los usuarios, los expertos y/o un grupo de evaluadores independientes. En este aspecto el SA implementado cumple con los parámetros que se plantean a continuación:

- El diseño y la implementación han sido llevados a cabo modularmente, porque se elaboró un cronograma por el cual se realizaron todas estas actividades y fueron hechas en un orden de

prioridad y complejidad.

- La interfaz de usuario cumple las especificaciones previstas pues se realizó siguiendo las pautas impuestas por el cliente y por las Guías de la interfaz humana de GNOME 2.2.2.
- Las facilidades de explicación son apropiadas para los potenciales usuarios del sistema. La Base de Conocimientos que posee el sistema de ayuda presenta problemas comunes con sus soluciones explicadas para que los usuarios no expertos puedan entenderlas y aplicarlas.
- El mantenimiento del sistema es posible hasta el grado especificado. Pues este cuenta con un sistema para “Gestionar información del Sistema de Ayuda” que permite gestionarlo.

Verificación de los mecanismos de inferencia.

En este aspecto el motor de inferencia utilizado por el SA fue el de la herramienta SWI-Prolog, por lo que no es necesario que se verifiquen los mecanismos de inferencia, atendiendo a que el motor de inferencia realiza esta verificación intrínsecamente.

Verificación de la Base de Conocimientos.

Por su parte, la verificación de la base de conocimientos se basa, generalmente, en el concepto de anomalías; una anomalía es un uso poco común del esquema de representación del conocimiento, que puede ser considerado como un error potencial (existen anomalías que no constituyen errores y viceversa). La verificación de la base de conocimientos no asegura que las respuestas del sistema sean correctas, lo que afirma es que el sistema ha sido diseñado e implementado de forma correcta. Los aspectos que se suelen examinar a la hora de verificar una base de conocimientos son la **consistencia** y la **completitud**. En la tabla 3.1 se muestran una serie de pruebas que se realizan para comprobar que la base de conocimientos es consistente y completa.

Consistencia	Reglas redundantes	$p(x) \wedge q(x) \rightarrow r(x)$ $q(x) \wedge p(x) \rightarrow r(x)$
	Reglas conflictivas	$p(x) \wedge q(x) \rightarrow r(x)$ $p(x) \wedge q(x) \rightarrow \neg r(x)$
	Reglas englobadas en otras	$p(x) \wedge q(x) \rightarrow r(x)$ $p(x) \rightarrow r(x)$
	Reglas circulares	$p(x) \rightarrow q(x)$ $q(x) \rightarrow r(x)$ $r(x) \rightarrow p(x)$
	Condiciones innecesarias	IF $p(x) \wedge q(x) \rightarrow r(x)$ $p(x) \wedge \neg q(x) \rightarrow r(x)$

Compleitud	Valores referenciados no de atributos	Ocorre cuando algunos valores, del conjunto de posibles valores de un atributo, no son cubiertos por la premisa de ninguna regla.
	Valores ilegales de atributos	Una regla referencia valores de atributos que no están incluidos en el conjunto de valores válidos para ese atributo.
	Reglas inalcanzables	$p(x) \rightarrow r(x)$ $p(x)$ no aparece como conclusión de otra regla ni puede obtenerse del exterior (razonamiento progresivo).
	Reglas sin salida	$p(x) \wedge q(x) \rightarrow r(x)$ $r(x)$ no es una conclusión final y no aparece en la premisa de ninguna regla (razonamiento progresivo).

Tabla 3.1 Verificación de la consistencia y la completitud en bases de conocimientos [35].

En la revisión realizada a las reglas de la base de conocimientos, en cuanto a la consistencia se determinó que no existían reglas redundantes, ni conflictivas, ni englobadas en otras reglas, ni circulares, ni condiciones IF innecesarias, ni reglas negativas. Además en el momento de adicionar una regla en el SA se realiza una validación que verifica que al sistema no se le inserten reglas que no cumplan con alguno de estos aspectos mencionados anteriormente. En cuanto a la completitud de las reglas, el SA implementado no presenta valores no referenciados, ni valores ilegales de atributos, ni reglas inalcanzables y sin salida. Lo anterior se demuestra pues las reglas para ser adicionadas dependen de que existan problemas con su solución en la base de conocimientos.

Conclusiones generales

Con el desarrollo del presente trabajo investigativo, se arribó a las siguientes conclusiones:

- El estudio de los SA para entornos informáticos, sus características, tipos, así como lenguajes y herramientas para su implementación; permitió identificar como solución al problema planteado la implementación de un SA del tipo Sistemas de Ayuda Inteligentes.
- El SA está compuesto por el sistema cliente “Ayuda y Soporte” y el sistema de gestión “Gestionar información del Sistema de Ayuda”. El sistema cliente permite al usuario resolver los problemas más frecuentes que presente con Nova Escritorio, mediante las soluciones que se gestionan a través del sistema de gestión.
- El SA aporta beneficios a sus usuarios y a la migración a aplicaciones de código abierto que experimenta Cuba, como parte del proceso de informatización de la sociedad y elimina la dependencia de la red para resolver sus problemas.
- A través del sistema de gestión (Gestionar información del Sistema de Ayuda) se podrá enriquecer la base de datos del SA, que además por su estructura permite ser actualizada desde el repositorio de Nova.
- Atendiendo a la poca documentación relacionada con los Sistemas de Ayuda Inteligentes, la solución propuesta constituye un avance en el desarrollo de la informática en Cuba.
- El SA puede ser utilizado para otras aplicaciones que necesiten una ayuda informatizada, partiendo de que su estructura es adaptable a diferentes contextos.

Recomendaciones

A partir del estudio realizado en la presente investigación se hacen las siguientes recomendaciones:

- Adicionarle al SA, la capacidad de realizar directamente las acciones sobre el sistema operativo, partiendo de la descripción de los pasos que se proponen en la solución del problema.
- Pasar el SA de pasivo a activo.
- El SA utilice la documentación del “*man*” para brindar más información al usuario.
- Que el SA sea utilizado en próximas ediciones y versiones de Nova.
- Mantener el seguimiento del SA por parte de los expertos del proyecto Nova.

Referencias bibliográficas

- [1] FERNÁNDEZ MANJÓN, Balastar. *Desarrollo de sistemas de ayuda inteligente mediante integración de tecnologías y reutilización de información*: Tesis doctoral. Departamento de Informática y Automática de la Universidad Complutense de Madrid. Madrid, 2001. p 85.
- [2] VIDAL LEDO, María. *Alfabetización Digital e Información de la Sociedad. Un reto para el presente*, Revista Cubana de Informática Médica (RCIM). [Citado 24 septiembre 2011]. Disponible en:<http://www.cecam.sld.cu/pages/rcim/revista_9/articulos_hm/alfabetizdigital.htm>.
- [3] PIERRA FUENTES, Allan. *CONCEPTUALIZACIÓN Y REESTRUCTURACIÓN ESTRATÉGICA DE LA DISTRIBUCIÓN CUBANA DE GNU/LINUX "NOVA"*: Tesis de Maestría. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2011. p 57-62.
- [4] LAGO HENRÍQUEZ, Emilio y CONCEPCIÓN PEÑA, Yusdel Juan. *Aplicación Web para brindar soporte a Nova: NovaDesk*. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2009. p 110. Disponible en:<http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_2426_09>.
- [5] TAMAYO ORO, Leanet y PAVÓN BERNAL, Yanet. *Herramienta para el Soporte de los Servicios NovaDesk 2.0*. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2010. p 103. Disponible en:<http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_03502_10>.
- [6] SÁNCHEZ ORTIZ, Susana y PÉREZ BENITEZ, Alfredo. *Sistema Experto para la gestión de la Base de Conocimientos de NovaDesk*. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2011. p 122. Disponible en:<http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_05082_11>.
- [7] HERNÁNDEZ LEÓN, Rolando Alfredo y COELLO GONZÁLEZ, Sayda. *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA*. Ciudad de La Habana, Editorial Universitaria Cubana, 2011. p 82-86. ISBN 978-959-16- 1307-3.
- [8] HERNÁNDEZ LEÓN, Rolando Alfredo y COELLO GONZÁLEZ, Sayda. *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA*. Ciudad de La Habana, Editorial Universitaria Cubana, 2011. p 86-92. ISBN 978-959-16- 1307-3.
- [9] FERNÁNDEZ MANJÓN, Balastar. *Desarrollo de sistemas de ayuda inteligente mediante integración de tecnologías y reutilización de información*: Tesis doctoral. Departamento de Informática y Automática de la Universidad Complutense de Madrid. Madrid, 2001. p 26.
- [10] SISTEMA. EN: Diccionario de Informática e Internet de Microsoft, *Cuarta edición*. [s.], McGraw-Hill/Interamericana, 2001. p 540. ISBN 84-481-2893-1.
- [11] AYUDA. EN: Diccionario de Informática e Internet de Microsoft, *Cuarta edición*. [s.], McGraw-

- Hill/Interamericana, 2001. p 540. ISBN 84-481-2893-1.
- [12] KEARSLEY, Greg. *Online Help Systems: Design and Implementation*. New Jersey, Ablex Publishing Corporation, 1988. p 56.
- [13] BELÉN GIL, Ana, y JOSÉ GARCÍA, Francisco. *Sistemas de apoyo en línea al usuario* [en línea], [Citado 19 octubre 2011]. Disponible en: <www.aipo.es/libro/doc/14Apoyo.doc>.
- [14] LAURO SOTO, Ensenada. *Sistemas De Apoyo Toma De Decisiones* [en línea], [Citado 21 octubre 2011]. Disponible en: <<http://www.mitecnologico.com/Main/SistemasDeApoyoTomaDeDecisiones>>.
- [15] SALGUEIRO, Fernando [et al.]. *Nuevo enfoque metodológico para el diseño de los sistemas tutores*, Revista de Informática Educativa y Medios Audiovisuales, 2005, 2, p 25-32.
- [16] FERNÁNDEZ MANJÓN, Balastar. *Sistemas de Ayuda Inteligente para Entornos Informáticos Complejos*, Revista Iberoamericana de Inteligencia Artificial, 2001, No 12, p 4. ISSN: 1137-3601.
- [17] FERNÁNDEZ MANJÓN, Balastar. *Sistemas de Ayuda Inteligente para Entornos Informáticos Complejos*, Revista Iberoamericana de Inteligencia Artificial, 2001, No 12, p 3-4. ISSN: 1137-3601.
- [18] FERNÁNDEZ MANJÓN, Balastar. *Desarrollo de sistemas de ayuda inteligente mediante integración de tecnologías y reutilización de información*: Tesis doctoral. Departamento de Informática y Automática de la Universidad Complutense de Madrid. Madrid, 2001. p 115.
- [19] *El primer sistema de ayuda a la decisión de diagnóstico de tumores cerebrales basado en código libre*. *Gaceta Tecnológica Espiral Tecnológica*, Abril 2008, p 13.
- [20] ÁLVAREZ GONZÁLEZ, Michael. *Sistema de Información Geográfico para el Análisis y Control de las Ayudas a la Navegación en La Bahía de Nuevitas (SAYNAV)* [en línea]. 2008. [Citado 24 octubre 2011]. Disponible en: <http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1517>.
- [21] FERNÁNDEZ CRESPO, Francisco [et al.]. *Sistema informático de apoyo al Diagnóstico Empresarial* [en línea], *Revista Avanzada Científica*, 2006, 9. Disponible en: <<http://avanzada.idict.cu/index.php/avanzada/article/download/149/157>>.
- [22] CANOVA RAMÍREZ, Dayana y PÉREZ DOPASO, Saidel. *Diseño e implementación del Sistema de Ayuda Médica para la Atención de las Dislipoproteinemias versión 2.0*. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2009. p 15. Disponible en: <http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_2505_09>.
- [23] ENOA PAYÉS, Leyanis y RODRÍGUEZ RODRÍGUEZ, Anisleidy. *Ayuda en línea para el sistema SENTAI de la Corporación de Importadores y Exportadores de Cuba (Cimex)*. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2009. p 7. Disponible en: <http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_2256_09>.

- [24] Sistema operativo. EN: *Conferencia 1. Arquitectura de los Sistemas Operativos Modernos*. Universidad de las Ciencias Informáticas. 2012. p 3. Disponible en: <<http://eva.uci.cu/course/view.php?id=146>>.
- [25] Ranking Sistemas Operativos Enero 2011 [en línea], [Citado 2 noviembre 2011]. Disponible en: <http://www.desarrolloweb.com/de_interes/ranking-sistemas-operativos-enero-2011-4783.html>.
- [26] *El lenguaje C++* [en línea], [Citado 2 noviembre 2011]. Disponible en: <http://www.zator.com/Cpp/E1_2.htm>.
- [27] *Lenguajes de programación, programación Java* [en línea], [Citado 3 noviembre 2011]. Disponible en: <<http://www.lenguajes-de-programacion.com/programacion-java.shtml>>.
- [28] *Lenguaje de programación Python* [en línea], [Citado 3 noviembre 2011]. Disponible en: <<http://programandoideas.com/el-lenguaje-de-programacion-python/>>.
- [29] PRESSMAN, Roger S. *Ingeniería del software: un enfoque práctico - 4a ed*. Madrid, McGraw-Hill, 1998. p 240-242. ISBN: 84-481-1186-9.
- [30] DÍAZ PASTRANA, Dainerys y PÉREZ PÉREZ, Joisel. *Modelo para la ayuda en la toma de decisiones relativas a estilos y patrones arquitectónicos*. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2008. p 44.
- [31] PEP 8 -- Style Guide for Python Code [en línea]. [Citado 12 abril 2012]. Disponible en: <<http://www.python.org/dev/peps/pep-0008/>>.
- [32] GLADYS MARSI, Peñalver Romero. *MA-GMPUR2 Metodología ágil para proyectos de software libre*. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2008. Disponible en: <https://repositorio.geitel.prod.uci.cu/svn/simays/SXP_metodologia_de_desarrollo/v0.2%20SXP/Documentos/>
- [33] FERNÁNDEZ MANJÓN, Balastar. *Desarrollo de sistemas de ayuda inteligente mediante integración de tecnologías y reutilización de información*: Tesis doctoral. Departamento de Informática y Automática de la Universidad Complutense de Madrid. Madrid, 2001. p 56-57.
- [34] FERNÁNDEZ MANJÓN, Balastar. *Desarrollo de sistemas de ayuda inteligente mediante integración de tecnologías y reutilización de información*: Tesis doctoral. Departamento de Informática y Automática de la Universidad Complutense de Madrid. Madrid, 2001. p 59-62.
- [35] MORET BONILLO, Vicente. *VALIDACIÓN Y USABILIDAD DE SISTEMAS INFORMÁTICOS*. 2005. p 30-33.

Bibliografía consultada

- ALLEN, Downey. *Aprenda a Pensar Como un Programador con Python*. 2002. ISBN 0-9716775-0-6.
- ÁLVAREZ GONZÁLEZ, Michael. *Sistema de Información Geográfico para el Análisis y Control de las Ayudas a la Navegación en La Bahía de Nuevitas (SAYNAV)* [en línea]. 2008. [Citado 24 octubre 2011]. Disponible en: <http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1517>.
- AYUDA. EN: Diccionario de Informática e Internet de Microsoft, *Cuarta edición*. [s.], McGraw-Hill/Interamericana, 2001. p 540. ISBN 84-481-2893-1.
- BELÉN GIL, Ana, y JOSÉ GARCÍA, Francisco. *Sistemas de apoyo en línea al usuario* [en línea], [Citado 19 octubre 2011]. Disponible en: <www.aipo.es/libro/doc/14Apoyo.doc>.
- *Bienvenido a NetBeans, Portal del IDE Java de Código Abierto* [en línea]. [Citado 4 noviembre 2011]. Disponible en: <http://netbeans.org/index_es.html>.
- CANOVA RAMÍREZ, Dayana y PÉREZ DOPASO, Saidel. *Diseño e implementación del Sistema de Ayuda Médica para la Atención de las Dislipoproteinemias versión 2.0*. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2009. p 15. Disponible en: <http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_2505_09>.
- *Design Patterns. Elements of Reusable Object-Oriented Software* - Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides - Addison Wesley (GoF- Gang of Four), ISBN 0-201-63361-2, 2005.
- DÍAZ PASTRANA, Dainerys y PÉREZ PÉREZ, Joisel. *Modelo para la ayuda en la toma de decisiones relativas a estilos y patrones arquitectónicos*. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2008. p 44.
- *El lenguaje C++* [en línea], [Citado 2 noviembre 2011]. Disponible en: <http://www.zator.com/Cpp/E1_2.htm>.
- *El primer sistema de ayuda a la decisión de diagnóstico de tumores cerebrales basado en código libre*. Gaceta Tecnológica Espiral Tecnológica, Abril 2008. p 13.
- *El primer sistema de ayuda a la decisión de diagnóstico de tumores cerebrales basado en código libre*. Gaceta Tecnológica Espiral Tecnológica, Abril 2008, p 13.
- ENOA PAYÉS, Leyanis y RODRÍGUEZ RODRÍGUEZ, Anisleidy. *Ayuda en línea para el sistema SENTAÍ de la Corporación de Importadores y Exportadores de Cuba (Cimex)*. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2009. p 7. Disponible en: <http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_2256_09>.

- FERNÁNDEZ CRESPO, Francisco [et al.]. *Sistema informático de apoyo al Diagnóstico Empresarial* [en línea], *Revista Avanzada Científica*, 2006, 9. Disponible en: <<http://avanzada.idict.cu/index.php/avanzada/article/download/149/157>>.
- FERNÁNDEZ MANJÓN, Balastar. *Desarrollo de sistemas de ayuda inteligente mediante integración de tecnologías y reutilización de información*: Tesis doctoral. Departamento de Informática y Automática de la Universidad Complutense de Madrid. Madrid, 2001. p 85.
- FERNÁNDEZ MANJÓN, Balastar. *Sistemas de Ayuda Inteligente para Entornos Informáticos Complejos*, *Revista Iberoamericana de Inteligencia Artificial*, 2001, No 12, ISSN: 1137-3601.
- FISCHER, G. *User Modeling: The Long and Winding Road*. Proceedings of the Seventh International Conference on User Modeling (UM99), Banff, Canada, 1999. p 20-24.
- FRAUSTO SOLIS, Juan [et al.]. *Un modelo de validación para la generación de explicaciones en un asistente inteligente para el entrenamiento de operadores*. 6to Congreso Internacional en Innovación y Desarrollo Tecnológico, CIINDET 2008.
- FUENTE DE LA TORRE LLORENTE, César y ZORRILLA, Unai. *Guía de arquitectura de N-Capas orientada al dominio con .NET*. España: Microsoft ibérica. 2010. ISBN 978-84-936696-3-8.
- GIMP [en línea]. [Citado 4 noviembre 2011]. Disponible en: <<http://gimp.es/>>.
- Glade - A User Interface Designer [en línea]. [Citado 16 enero 2012]. Disponible en: <http://glade.gnome.org/>.
- GLADYS MARSÍ, Peñalver Romero. *MA-GMPR-UR2 Metodología ágil para proyectos de software libre*. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2008. Disponible en: <https://repositorio.geitel.prod.uci.cu/svn/simays/SXP_metodologia_de_desarrollo/v0.2%20SXP/Documentos/>
- GODINHO, Pedro. *CISTI (Iberian Conference on Information Systems & Technologies / Conferencia Ibérica de Sistemas de Tecnologías de Información) Proceedings*, 2011, p 428-433, 6p, 4 Diagrams; Language: Portuguese, EBSCO, Database: Computers & Applied Sciences Complete.
- GTK+2 Reference Manual [en línea]. [Citado 16 enero 2012]. Disponible en: <<http://developer.gnome.org/gtk/stable/>>.
- Guía Breve de CSS [en línea]. [Citado 16 enero 2012]. Disponible en: <<http://www.w3c.es/divulgacion/guiasbreves/hojasestilo>>.
- *Guía Breve de XHTML* [en línea]. [Citado 3 diciembre 2011]. Disponible en:

- <<http://www.w3c.es/Divulgacion/Guiasbreves/XHTML>>.
- HERNÁNDEZ LEÓN, Rolando Alfredo y COELLO GONZÁLEZ, Sayda. *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA*. Ciudad de La Habana, Editorial Universitaria Cubana, 2011. ISBN 978-959-16- 1307-3.
 - HERNÁNDEZ SAMPIERI, Roberto [et al.]. *Metodología de la Investigación: Cuarta edición*. The McGraw-Hill Companies. 2006. ISBN 970-10-5753-8.
 - HTML.COM [en línea]. [Citado 16 enero 2012]. Disponible en: <<http://www.html.com/>>.
 - isg2/Pruebas del Software. [Citado 6 mayo 2012]. Disponible en: <<http://isg2.pbworks.com/w/page/7624280/Pruebas%20del%20Software>>.
 - J, Carletti. *Ayuda informática para la memoria* [en línea]. [Citado 21 Octubre 2011], Disponible en: <<http://axxon.com.ar/not/151/c-1510151.htm>>.
 - *Jetlogs.org* » *Apache vs. Lighttpd on PHP* [en línea]. [Citado 4 diciembre 2011]. Disponible en: <<http://jetlogs.org/2007/08/23/apache-vs-lighttpd-on-php/>>.
 - KEARSLEY, Greg. *Online Help Systems: Design and Implementation*. New Jersey, Ablex Publishing Corporation, 1988. p 56.
 - LAGO HENRÍQUEZ, Emilio y CONCEPCIÓN PEÑA, Yusdel Juan. *Aplicación Web para brindar soporte a Nova: NovaDesk*. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2009. p 110. Disponible en:<http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_2426_09>.
 - LAURO SOTO, Ensenada. *Sistemas De Apoyo Toma De Decisiones* [en línea], [Citado 21 octubre 2011]. Disponible en: <<http://www.mitecnologico.com/Main/SistemasDeApoyoTomaDeDecisiones>>.
 - *Lenguaje de programación Python* [en línea], [Citado 3 noviembre 2011]. Disponible en: <<http://programandoideas.com/el-lenguaje-de-programacion-python/>>.
 - *Lenguajes de programación, programación Java* [en línea], [Citado 3 noviembre 2011]. Disponible en: <<http://www.lenguajes-de-programacion.com/programacion-java.shtml>>.
 - MORET BONILLO, Vicente. *VALIDACIÓN Y USABILIDAD DE SISTEMAS INFORMÁTICOS*. 2005. p 30-33.
 - PEP 8 -- Style Guide for Python Code [en línea]. [Citado 12 abril 2012]. Disponible en: <<http://www.python.org/dev/peps/pep-0008/>>.
 - PÉREZ MONTORO, Mario. *El Profesional de la Información*, jul/ago2010, Vol. 19 Issue 4, p 333-337, p 5, 4 Color Photographs; Language: Spanish; DOI: 10.3145/epi.2010.jul.01, EBSCO, Database: Computers & Applied Sciences.

- PIERRA FUENTES, Allan. *CONCEPTUALIZACIÓN Y REESTRUCTURACIÓN ESTRATÉGICA DE LA DISTRIBUCIÓN CUBANA DE GNU/LINUX “NOVA”*. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2011. p 57-62.
- PRESSMAN, Roger S. *Ingeniería del software: un enfoque práctico - 4a ed.* Madrid, McGraw-Hill, 1998. p 240-242. ISBN: 84-481-1186-9.
- Ranking Sistemas Operativos Enero 2011 [en línea], [Citado 2 noviembre 2011]. Disponible en: <http://www.desarrolloweb.com/de_interes/ranking-sistemas-operativos-enero-2011-4783.html>.
- SALGUEIRO, Fernando [et al.]. *Nuevo enfoque metodológico para el diseño de los sistemas tutores*, Revista de Informática Educativa y Medios Audiovisuales, 2005, 2, p 25-32.
- SÁNCHEZ ORTIZ, Susana y PÉREZ BENITEZ, Alfredo. *Sistema Experto para la gestión de la Base de Conocimientos de NovaDesk*. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2011. p 122. Disponible en: <http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_05082_11>.
- SIERRA, María. *Trabajando con Visual Paradigm for UML*. 2005. Disponible en: <<http://personales.unican.es/ruijzfr/is1/doc/lab/01/is1-p01-trans.pdf>>.
- Sistema operativo. EN: *Conferencia1. Arquitectura de los Sistemas Operativos Modernos*. Universidad de las Ciencias Informáticas. 2012. p 3. Disponible en: <<http://eva.uci.cu/course/view.php?id=146>>.
- SISTEMA. EN: *Diccionario de Informática e Internet de Microsoft, Cuarta edición*. [s.l.], McGraw-Hill/Interamericana, 2001. p 540. ISBN 84-481-2893-1.
- TAMAYO ORO, Leanet y PAVÓN BERNAL, Yanet. *Herramienta para el Soporte de los Servicios NovaDesk 2.0*. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2010. p 103. Disponible en: <http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_03502_10>.
- TANENBAUM, Andrew Stuart y WOODHULL, Albert S. *SISTEMAS OPERATIVOS Diseño e implementación*. Segunda edición. 1998. ISBN 970-17-0165-8.
- TANENBAUM, Andrew Stuart. *SISTEMAS OPERATIVOS MODERNOS*. Tercera edición. 2009. ISBN 0-13-638677-6.
- Tutorial de Prolog [en línea]. [Citado 16 enero 2012]. Disponible en: <<http://proton.ucting.udg.mx/tutorial/prolog/index.htm>>.
- VIDAL LEDO, María. *Alfabetización Digital e Información de la Sociedad. Un reto para el presente*, Revista Cubana de Informática Médica (RCIM). [Citado 24 septiembre 2011]. Disponible en: <http://www.cecarn.sld.cu/pages/rcim/revista_9/articulos_htm/alfabetizdigital.htm>.

Anexos

Anexo 1 : Principales preguntas que conformaron las entrevistas realizadas al cliente.

- ¿Cuáles son las tecnologías que sugiere utilizar para la implementación del SA?
- ¿Qué tipo de aplicación necesitan, web o de escritorio?
- ¿Qué categorías y subcategorías se pueden utilizar en el SA para la edición Nova Escritorio?
- ¿Cuáles son los problemas más comunes a los que se enfrenta un usuario cuando utiliza Nova Escritorio?
- ¿En qué categorías clasificaría cada uno de estos problemas?
- ¿Cómo se identifican estos problemas?
- ¿Qué solución se les daría a estos problemas?
- ¿Cuánta información debe tener la base de datos para ser aceptado el producto?

Anexo 2 : Datos del cliente/experto humano.

Ing. Daniel Hernández Bahr, nació el 14 de junio de 1986, es recién graduado en adiestramiento. Se desempeña como Jefe del Proyecto Nova Escritorio que se desarrolla en el Centro de *Software* Libre (CESOL). Tiene 5 años de experiencia en el desarrollo de aplicaciones libres para sistemas GNU/Linux, específicamente Nova.

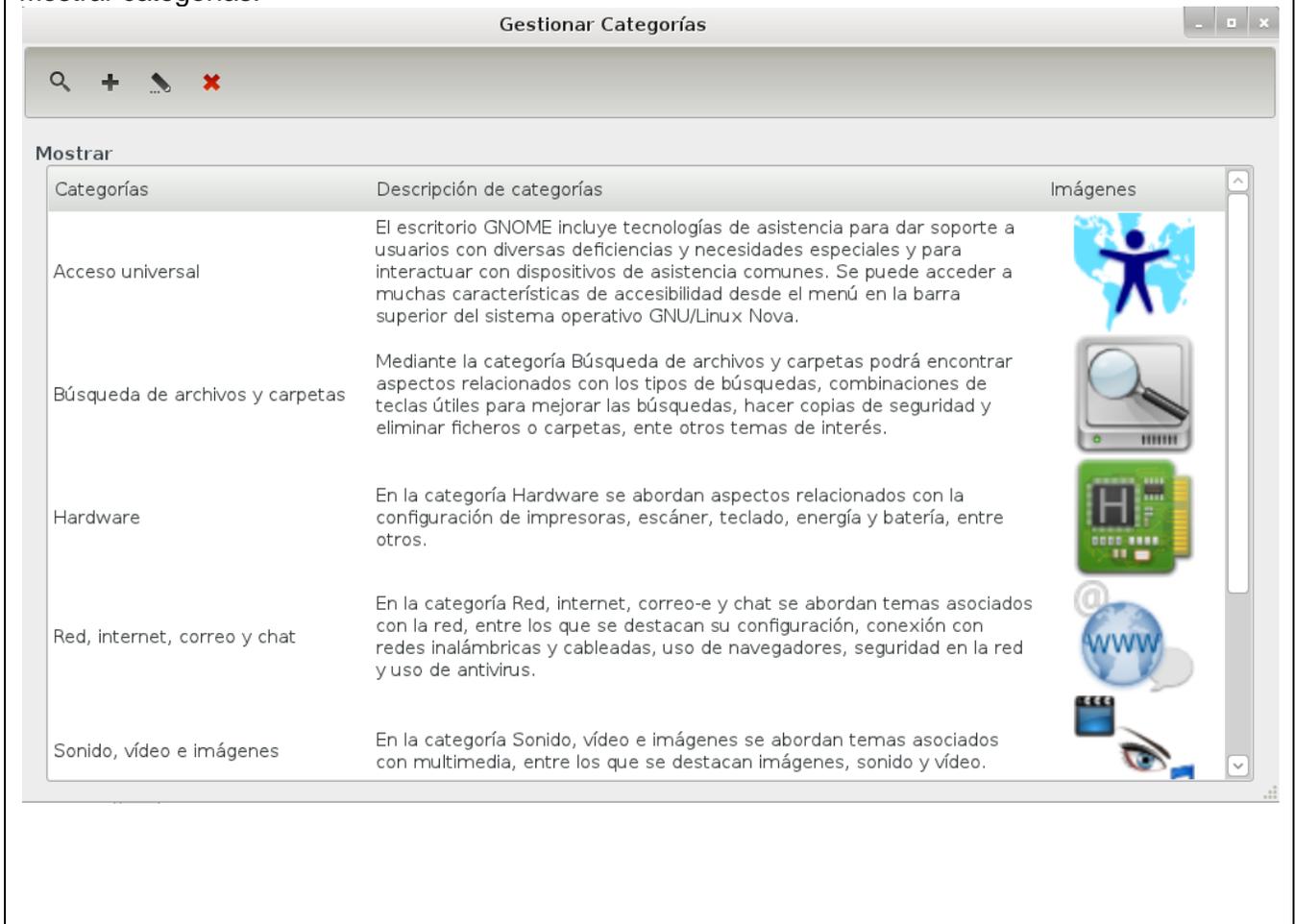
Anexo 3 : Historias de Usuario con sus respectivas tareas de ingeniería.

I. Gestionar categoría.

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Gestionar categoría
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoency Leyva Ochoa y José Gustavo Suárez Matilla	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: Debe permitir adicionar, editar, mostrar y eliminar categorías. Estas categorías serán seleccionadas teniendo en cuenta el criterio del cliente, las categorías de la aplicación Web NovaDesk 2.0 y las que usan otros SA.	
Observaciones: Primero se tienen que definir las categorías en un lenguaje natural para luego adicionarlas al SA.	
Prototipo de interfaz: Adicionar categorías:	



Mostrar categorías:



Editar categorías:

Eliminar categorías:

Tareas de Ingeniería

Tarea de Ingeniería	
Número Tarea: 1.1	Número Historia de Usuario: HU_1
Nombre Tarea: Definir categorías.	
Tipo de Tarea: Investigación	Puntos Estimados: 1
Fecha Inicio: 01/12/2011	Fecha Fin: 02/12/2011
Programador Responsable: Yoency Leyva Ochoa	
Descripción: Definir textualmente las categorías que conformarán el SA.	

Tarea de Ingeniería	
Número Tarea: 1.2	Número Historia de Usuario: HU_1
Nombre Tarea: Adicionar categoría.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 02/12/2011	Fecha Fin: 04/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita adicionar categorías al SA.	

Tarea de Ingeniería	
Número Tarea: 1.3	Número Historia de Usuario: HU_1
Nombre Tarea: Mostrar categorías.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 02/12/2011	Fecha Fin: 04/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita mostrar las categorías del SA.	

Tarea de Ingeniería	
Número Tarea: 1.4	Número Historia de Usuario: HU_1
Nombre Tarea: Editar categorías.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 06/12/2011	Fecha Fin: 08/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita editar las categorías del SA.	

Tarea de Ingeniería	
Número Tarea: 1.5	Número Historia de Usuario: HU_1
Nombre Tarea: Eliminar categorías.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 04/12/2011	Fecha Fin: 06/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita eliminar las categorías del SA.	

II. Gestionar subcategoría.

Historia de Usuario	
Número: HU_2	Nombre Historia de Usuario: Gestionar subcategoría.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoency Leyva Ochoa y José Gustavo Suárez Matilla	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El SA tendrá subcategorías compuestas por preguntas que permitirán identificar el posible problema que tiene el usuario. Estas subcategorías serán adicionadas, modificadas, mostradas y eliminadas.	
Observaciones: Primero tienen que ser adicionadas las categorías al SA.	

Prototipo de interfaz:

Adicionar subcategorías:

Mostrar subcategorías:

The screenshot shows a window titled "Gestionar Subcategoría" with a toolbar containing a search icon, a plus sign, a pencil, and a red X. Below the toolbar, the "Mostrar" section features a dropdown menu labeled "Seleccionar Categoría" and a table with two columns: "Subcategorías" and "Descripciones". The table is currently empty.

Editar subcategorías:

The screenshot shows the "Gestionar Subcategoría" window in "Editar" mode. It includes the same toolbar as the previous view. The form contains four input fields: "Seleccionar Categoría" (dropdown), "Seleccionar Subcategoría" (dropdown), "Nombre" (text), and "Descripción" (text area). At the bottom, there are two buttons: "Cancelar" (with a red prohibition icon) and "Aceptar" (with a checkmark icon).

Eliminar subcategorías:

The screenshot shows the "Gestionar Subcategoría" window in "Eliminar" mode. It features the same toolbar. The form contains two dropdown menus: "Seleccionar Categoría" and "Seleccionar Subcategoría". A "Cancelar" button with a red prohibition icon is located at the bottom right.

Tareas de Ingeniería

Tarea de Ingeniería	
Número Tarea: 2.1	Número Historia de Usuario: HU_2
Nombre Tarea: Definir subcategorías.	
Tipo de Tarea : Investigación	Puntos Estimados: 1
Fecha Inicio: 09/12/2011	Fecha Fin: 10/12/2011
Programador Responsable: Yoency Leyva Ochoa	
Descripción: Definir textualmente las subcategorías que conformarán el SA.	

Tarea de Ingeniería	
Número Tarea: 2.2	Número Historia de Usuario: HU_2
Nombre Tarea: Adicionar subcategorías.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 10/12/2011	Fecha Fin: 12/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita adicionar las subcategorías al SA.	

Tarea de Ingeniería	
Número Tarea: 2.3	Número Historia de Usuario: HU_2
Nombre Tarea: Mostrar subcategorías.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 10/12/2011	Fecha Fin: 12/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita mostrar las subcategorías del SA.	

Tarea de Ingeniería	
Número Tarea: 2.4	Número Historia de Usuario: HU_2
Nombre Tarea: Editar subcategorías.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 14/12/2011	Fecha Fin: 16/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita editar las subcategorías del SA.	

Tarea de Ingeniería	
Número Tarea: 2.5	Número Historia de Usuario: HU_2
Nombre Tarea: Eliminar subcategorías.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 12/12/2011	Fecha Fin: 14/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita eliminar las subcategorías del SA.	

III. Gestionar contenido (Problemas con su solución y Preguntas frecuentes).

Historia de Usuario	
Número: HU_3	Nombre Historia de Usuario: Gestionar contenido (Problemas con su Solución y preguntas frecuentes)
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoency Leyva Ochoa y José Gustavo Suárez Matilla	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1

Descripción: El SA contará con categorías y estas por subcategorías conformadas por preguntas que permitirán identificar el posible problema que tiene el usuario. Además se gestionarán las preguntas frecuentes que responden a los temas más usados por los usuarios, tanto los problemas con su solución como las preguntas frecuentes serán adicionados, modificados, mostrados y eliminados.

Observaciones:

Prototipo de interfaz:

Adicionar problema:

The screenshot shows a window titled "Gestionar Contenido (Problema con su solución)". The main area is titled "Adicionar". It contains the following elements:

- A search icon and three action icons (+, edit, delete) in the top left.
- Two dropdown menus: "Seleccionar Categoría" and "Seleccionar Subcategoría".
- Text input fields for "Nombre" and "Descripción".
- A section titled "Adicionar Solución" containing a table with two columns: "Descripción de la solución" and "Nombre de la imagen".
- At the bottom, four buttons: "+ Añadir", "Editar", "Eliminar", and "Cancelar".

Mostrar problema:

The screenshot shows a window titled "Gestionar Contenido (Problema con su solución)". The main area is titled "Mostrar". It contains the following elements:

- A search icon and three action icons (+, edit, delete) in the top left.
- Three dropdown menus: "Seleccionar categoría", "Seleccionar Subcategoría", and "Seleccionar Problema".
- A large empty rectangular area below the dropdowns, intended for displaying the selected problem content.

Eliminar problema:

Gestionar Contenido (Problema con su solución)

Eliminar

Seleccionar categoría

Seleccionar Subcategoría

Seleccionar Problema

✖ Eliminar

Adicionar preguntas frecuentes:

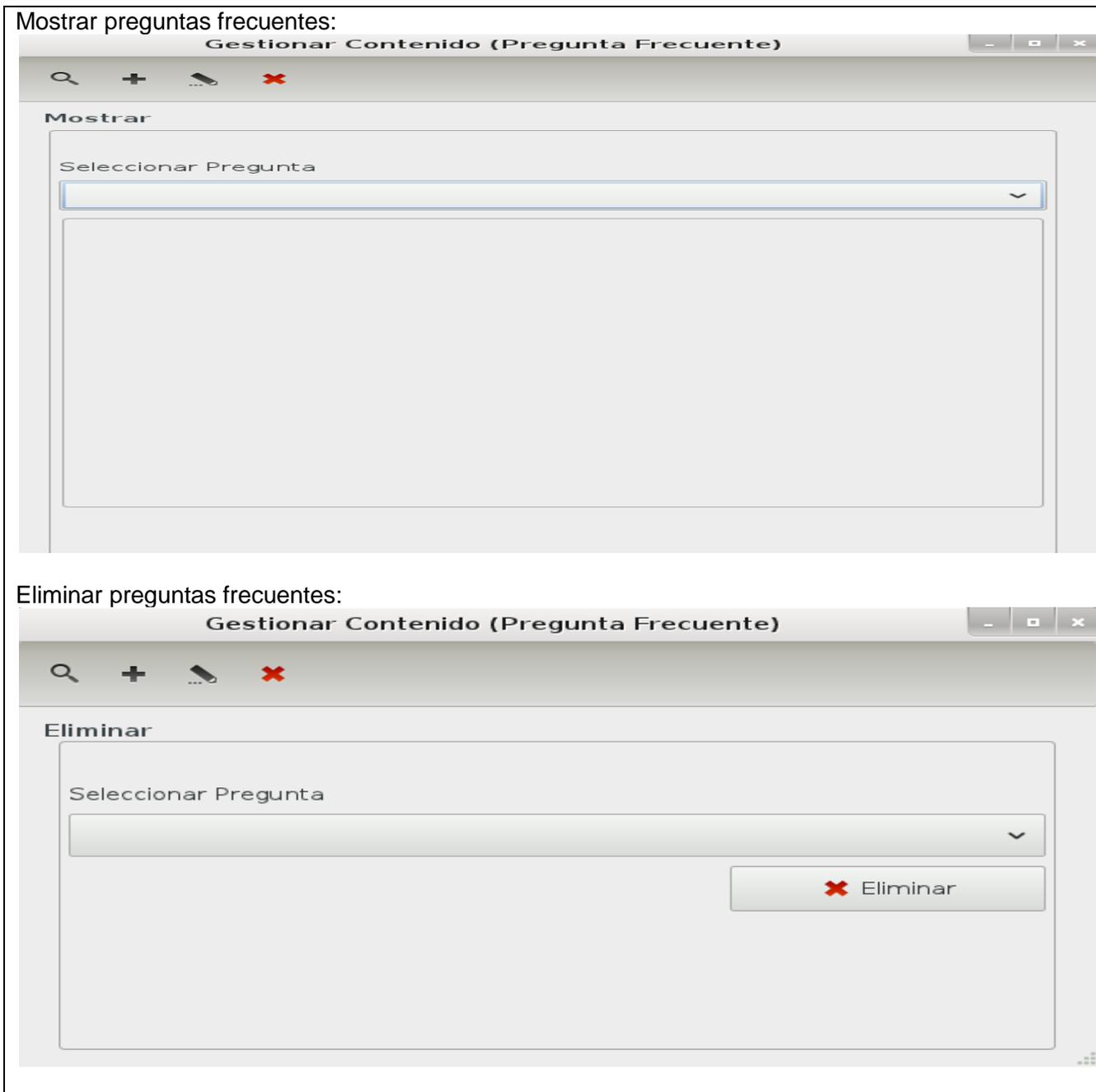
Gestionar Contenido (Pregunta Frecuente)

Adicionar

Nombre

Adicionar Solución

Descripción de la solución	Nombre de la imagen



Tareas de Ingeniería

Tarea de Ingeniería	
Número Tarea: 3.1	Número Historia de Usuario: HU_3
Nombre Tarea: Identificar principales problemas y su solución de la edición Nova Escritorio.	
Tipo de Tarea: Investigación	Puntos Estimados: 1
Fecha Inicio: 17/12/2011	Fecha Fin: 19/12/2011
Programador Responsable: Yoency Leyva Ochoa	
Descripción: Identificar los principales problemas que presenta la edición Nova Escritorio.	

Tarea de Ingeniería	
Número Tarea: 3.2	Número Historia de Usuario: HU_3
Nombre Tarea: Identificar solución a los problemas.	
Tipo de Tarea : Investigación	Puntos Estimados: 1
Fecha Inicio: 17/12/2011	Fecha Fin: 20/12/2011
Programador Responsable: Yoency Leyva Ochoa	
Descripción: Identificar la solución a los principales problemas que presenta la edición Nova Escritorio.	

Tarea de Ingeniería	
Número Tarea: 3.3	Número Historia de Usuario: HU_3
Nombre Tarea: Adicionar problemas con su solución.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 18/12/2011	Fecha Fin: 20/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita adicionar los problemas con su solución al SA.	

Tarea de Ingeniería	
Número Tarea: 3.4	Número Historia de Usuario: HU_3
Nombre Tarea: Mostrar los problemas con su solución.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 18/12/2011	Fecha Fin: 20/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita mostrar los problemas con su solución del SA.	

Tarea de Ingeniería	
Número Tarea: 3.5	Número Historia de Usuario: HU_3
Nombre Tarea: Editar problemas con su solución.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 18/12/2011	Fecha Fin: 21/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita editar los problemas con su solución al SA.	

Tarea de Ingeniería	
Número Tarea: 3.6	Número Historia de Usuario: HU_3
Nombre Tarea: Eliminar problemas con su solución.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 20/12/2011	Fecha Fin: 22/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita eliminar los problemas con su solución al SA.	

Tarea de Ingeniería	
Número Tarea: 3.7	Número Historia de Usuario: HU_3
Nombre Tarea: Identificar preguntas frecuentes.	
Tipo de Tarea : Investigación	Puntos Estimados: 1
Fecha Inicio: 20/12/2011	Fecha Fin: 26/12/2011
Programador Responsable: Yoency Leyva Ochoa	

Descripción: Identificar las preguntas frecuentes atendiendo al criterio del cliente.	
Tarea de Ingeniería	
Número Tarea: 3.8	Número Historia de Usuario: HU_3
Nombre Tarea: Adicionar preguntas frecuentes.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22/12/2011	Fecha Fin: 24/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita Adicionar preguntas frecuentes al SA.	
Tarea de Ingeniería	
Número Tarea: 3.9	Número Historia de Usuario: HU_3
Nombre Tarea: Mostrar preguntas frecuentes.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22/12/2011	Fecha Fin: 24/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita mostrar las preguntas frecuentes del SA.	
Tarea de Ingeniería	
Número Tarea: 3.10	Número Historia de Usuario: HU_3
Nombre Tarea: Editar las preguntas frecuentes.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 24/12/2011	Fecha Fin: 27/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita editar los problemas con su solución al SA.	
Tarea de Ingeniería	
Número Tarea: 3.11	Número Historia de Usuario: HU_3
Nombre Tarea: Eliminar las preguntas frecuentes.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 28/12/2011	Fecha Fin: 29/12/2011
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita eliminar las preguntas frecuentes SA.	

IV. Gestionar Preguntas.

Historia de Usuario	
Número: HU_4	Nombre Historia de Usuario: Gestionar preguntas
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoency Leyva Ochoa y José Gustavo Suárez Matilla	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: Debe permitir adicionar, editar, eliminar y mostrar las preguntas que forman parte del SA. Estas preguntas constituyen la base para que el sistema cliente (Ayuda y Soporte) pueda determinar el problema al que se está enfrentando el usuario en ese momento.	
Observaciones: Para Adicionar preguntas, debe existir al menos una categoría con subcategorías.	

Prototipo de interfaz:

Adicionar preguntas:

The screenshot shows a window titled "Gestionar Preguntas" with a toolbar containing a search icon, a plus sign, a pencil, and a red X. The main area is labeled "Adicionar" and contains the following elements:

- A dropdown menu labeled "Seleccionar Categoría".
- A dropdown menu labeled "Seleccionar Subcategoría".
- A text input field labeled "Pregunta".
- A list area labeled "Lista de preguntas" with a sub-header "Preguntas" and an empty list.
- A button labeled "+ Añadir" at the bottom right.

Mostrar preguntas:

The screenshot shows a window titled "Gestionar Preguntas" with a toolbar containing a search icon, a plus sign, a pencil, and a red X. The main area is labeled "Mostrar" and contains the following elements:

- A dropdown menu labeled "Seleccionar Categoría".
- A dropdown menu labeled "Seleccionar Subcategoría".
- A list area labeled "Preguntas" with an empty list.
- A small grid icon in the bottom right corner.

Editar preguntas:

Eliminar preguntas:

Tareas de Ingeniería

Tarea de Ingeniería	
Número Tarea: 4.1	Número Historia de Usuario: HU_4
Nombre Tarea: Definir las preguntas.	
Tipo de Tarea : Investigación	Puntos Estimados: 1
Fecha Inicio: 02/01/2012	Fecha Fin: 04/01/2012
Programador Responsable: Yoency Leyva Ochoa	
Descripción: Definir textualmente las preguntas que conformarán el SA.	
Tarea de Ingeniería	
Número Tarea: 4.2	Número Historia de Usuario: HU_4

Nombre Tarea: Clasificar y agrupar las preguntas del SA por categoría y subcategoría.	
Tipo de Tarea : Investigación	Puntos Estimados: 1
Fecha Inicio: 02/01/2012	Fecha Fin: 05/01/2012
Programador Responsable: Yoency Leyva Ochoa	
Descripción: Clasificar y agrupar las preguntas que conformarán el SA de acuerdo a la categoría y subcategoría.	

Tarea de Ingeniería	
Número Tarea: 4.3	Número Historia de Usuario: HU_4
Nombre Tarea: Adicionar preguntas.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 06/01/2012	Fecha Fin: 09/01/2012
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita adicionar preguntas al SA.	

Tarea de Ingeniería	
Número Tarea: 4.4	Número Historia de Usuario: HU_4
Nombre Tarea: Mostrar preguntas.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 10/01/2012	Fecha Fin: 13/01/2012
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita mostrar las preguntas del SA.	

Tarea de Ingeniería	
Número Tarea: 4.5	Número Historia de Usuario: HU_4
Nombre Tarea: Eliminar preguntas.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 13/01/2012	Fecha Fin: 15/01/2012
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita eliminar las preguntas del SA.	

Tarea de Ingeniería	
Número Tarea: 4.6	Número Historia de Usuario: HU_4
Nombre Tarea: Editar preguntas.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 13/01/2012	Fecha Fin: 15/01/2012
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita editar las preguntas del SA.	

V. Gestionar reglas.

Historia de Usuario	
Número: HU_5	Nombre Historia de Usuario: Gestionar reglas.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoency Leyva Ochoa	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: Debe permitir adicionar, mostrar, editar y eliminar reglas del SA.	
Observaciones: Para Adicionar reglas tienen que existir categorías, subcategorías, problemas con su solución y preguntas.	

Prototipo de interfaz:

Adicionar reglas:

The screenshot shows a window titled "Gestionar Regla" with a toolbar containing search, add, edit, and delete icons. The main area is titled "Adicionar" and contains three dropdown menus for "Seleccionar Categoría", "Seleccionar Subcategoría", and "Seleccionar Problema". Below these is a section "Lista de preguntas" with two empty list boxes: "Preguntas" and "Preguntas Adicionadas". At the bottom, there are "Cancelar" and "Aceptar" buttons.

Mostrar reglas:

The screenshot shows the same "Gestionar Regla" window, but the main area is titled "Mostrar". The "Seleccionar Categoría" dropdown is set to "Red, internet, correo y chat" and "Seleccionar Subcategoría" is set to "Conexiones cableadas". A large text area labeled "Regla" contains the following text: "conectarse_a_una_vpn:-redcomacoma_internetcomacoma_correo_y_chat,conexiones_cableadas,ct" and "configurar_la_red_cableada:-redcomacoma_internetcomacoma_correo_y_chat,conexiones_cableada". A scrollbar is visible at the bottom of the text area.

Editar reglas:

Eliminar reglas:

Tareas de Ingeniería

Tarea de Ingeniería	
Número Tarea: 5.1	Número Historia de Usuario: HU_5
Nombre Tarea: Adicionar reglas.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 16/01/2012	Fecha Fin: 18/01/2012
Programador Responsable: Yoency Leyva Ochoa	
Descripción: Implementar la funcionalidad que permita adicionar reglas al SA.	

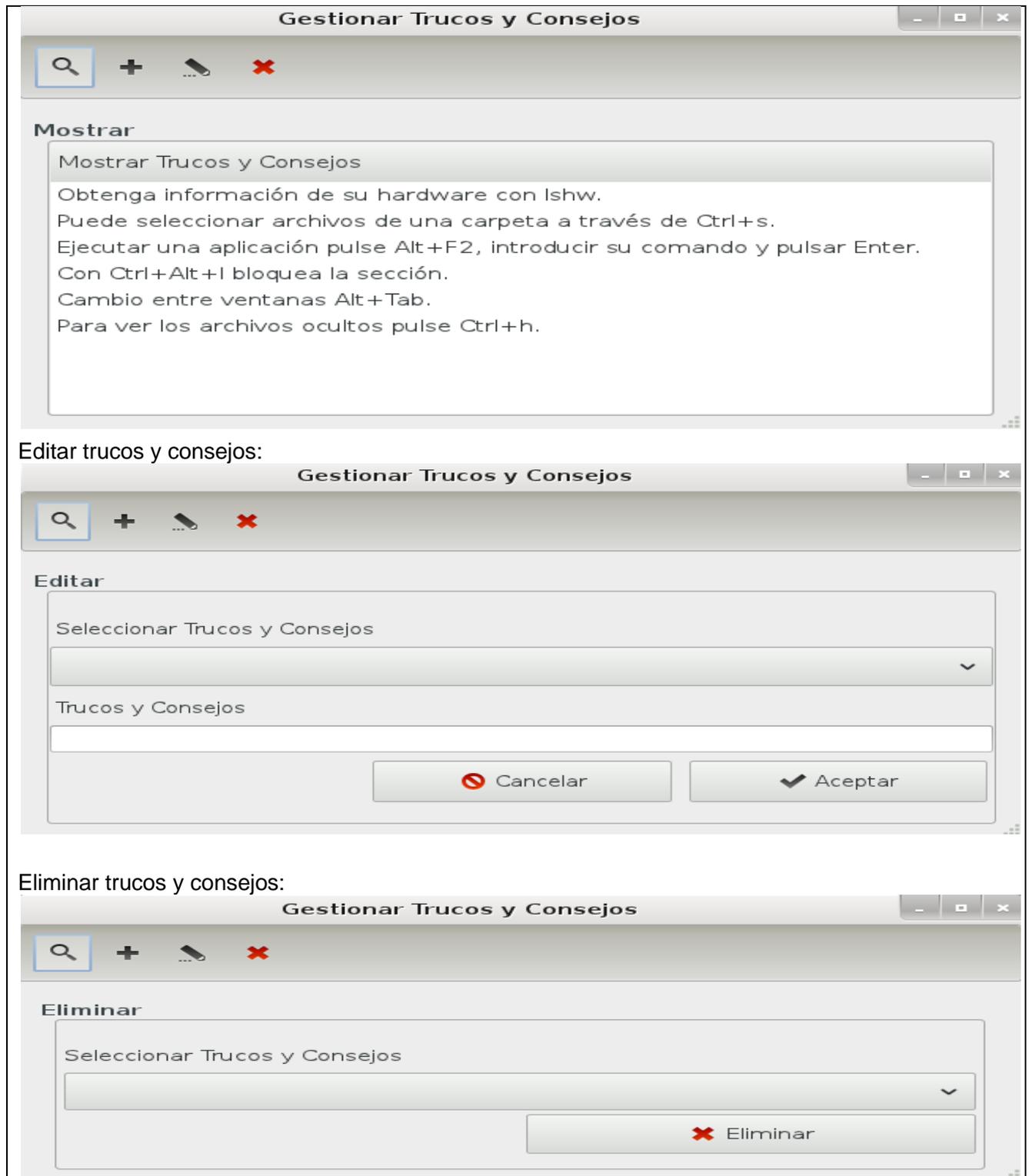
Tarea de Ingeniería	
Número Tarea: 5.2	Número Historia de Usuario: HU_5
Nombre Tarea: Mostrar reglas.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 16/01/2012	Fecha Fin: 18/01/2012
Programador Responsable: Yoency Leyva Ochoa	
Descripción: Implementar la funcionalidad que permita mostrar las reglas del SA.	

Tarea de Ingeniería	
Número Tarea: 5.3	Número Historia de Usuario: HU_5
Nombre Tarea: Eliminar reglas.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 18/01/2012	Fecha Fin: 19/01/2012
Programador Responsable: Yoency Leyva Ochoa	
Descripción: Implementar la funcionalidad que permita eliminar reglas al SA.	

Tarea de Ingeniería	
Número Tarea: 5.4	Número Historia de Usuario: HU_5
Nombre Tarea: Editar reglas.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 18/01/2012	Fecha Fin: 20/01/2012
Programador Responsable: Yoency Leyva Ochoa	
Descripción: Implementar la funcionalidad que permita editar reglas al SA.	

VI. Gestionar trucos y consejos

Historia de Usuario	
Número: HU_6	Nombre Historia de Usuario: Gestionar trucos y consejos.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoency Leyva Ochoa	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: Debe permitir adicionar, mostrar, editar y eliminar trucos y consejos del SA.	
Observaciones: Estos trucos y consejos son pequeñas textos informativos útiles al usuario.	
Prototipo de interfaz:	
Adicionar trucos y consejos:	
	
Mostrar trucos y consejos:	



Tareas de Ingeniería

Tarea de Ingeniería	
Número Tarea: 6.1	Número Historia de Usuario: HU_6
Nombre Tarea: Adicionar trucos y consejos.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 20/01/2012	Fecha Fin: 22/01/2012
Programador Responsable: Yoency Leyva Ochoa	
Descripción: Implementar la funcionalidad que permita adicionar trucos y consejos al SA.	

Tarea de Ingeniería	
Número Tarea: 6.2	Número Historia de Usuario: HU_6
Nombre Tarea: Mostrar trucos y consejos.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22/01/2012	Fecha Fin: 24/01/2012
Programador Responsable: Yoency Leyva Ochoa	
Descripción: Implementar la funcionalidad que permita mostrar los trucos y consejos del SA.	

Tarea de Ingeniería	
Número Tarea: 6.3	Número Historia de Usuario: HU_6
Nombre Tarea: Eliminar trucos y consejos.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 24/01/2012	Fecha Fin: 26/01/2012
Programador Responsable: Yoency Leyva Ochoa	
Descripción: Implementar la funcionalidad que permita eliminar trucos y consejos del SA.	

Tarea de Ingeniería	
Número Tarea: 6.4	Número Historia de Usuario: HU_6
Nombre Tarea: Editar trucos y consejos.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 26/01/2012	Fecha Fin: 30/01/2012
Programador Responsable: Yoency Leyva Ochoa	
Descripción: Implementar la funcionalidad que permita editar trucos y consejos SA.	

VII. Identificar el problema mediante la selección del usuario.

Historia de Usuario	
Número: HU_7	Nombre Historia de Usuario: Identificar el problema mediante la selección del usuario.
Modificación de Historia de Usuario Número: ninguna	
Usuario: José Gustavo Suárez Matilla	Iteración Asignada: 4
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: La presente HU tiene como objetivo identificar el problema que tiene el usuario, teniendo en cuenta la selección del usuario de las preguntas que pertenecen.	
Observaciones: El usuario debe seleccionar la categoría, subcategoría y la(s) pregunta(s) para que el sistema de ayuda pueda identificar el problema.	
Prototipo de interfaz:	

Tareas de Ingeniería

Tarea de Ingeniería	
Número Tarea: 7.1	Número Historia de Usuario: HU_7

Nombre Tarea: Identificar el problema relacionado con la selección del usuario.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 01/02/2012	Fecha Fin: 03/02/2012
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita identificar el problema relacionado con la selección del usuario.	

VIII. Buscar información.

Historia de Usuario	
Número: HU_8	Nombre Historia de Usuario: Buscar información.
Modificación de Historia de Usuario Número: ninguna	
Usuario: José Gustavo Suárez Matilla	Iteración Asignada: 4
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: Permitir buscar información relacionada con algún parámetro de búsqueda introducido por el usuario en un campo disponible para ello.	
Observaciones: El parámetro de búsqueda debe estar escrito correctamente.	
Prototipo de interfaz:	
	

Tareas de Ingeniería

Tarea de Ingeniería	
Número Tarea: 8.1	Número Historia de Usuario: HU_8
Nombre Tarea: Buscar problemas.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 04/02/2012	Fecha Fin: 10/02/2012
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita buscar información en el SA.	

IX. Exportar a formato pdf.

Historia de Usuario	
Número: HU_9	Nombre Historia de Usuario: Exportar a formato pdf.
Modificación de Historia de Usuario Número: ninguna	
Usuario: José Gustavo Suárez Matilla	Iteración Asignada: 4
Prioridad en Negocio: Alta	Puntos Estimados: 1

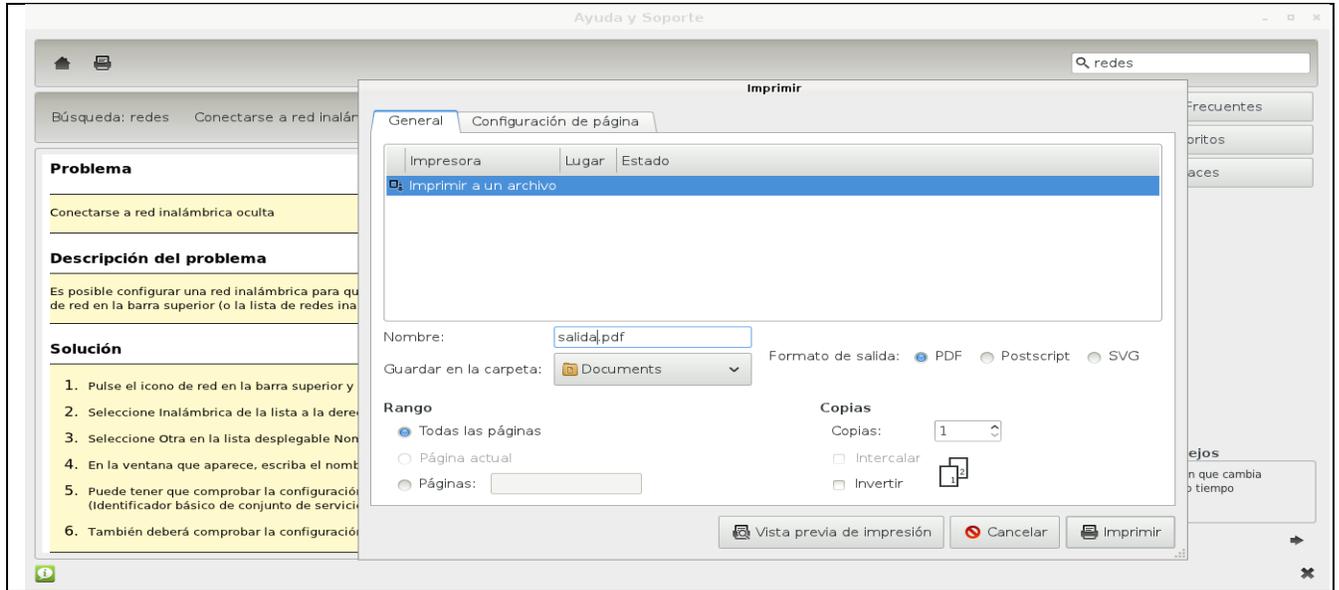
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: Debe permitir exportar a formato pdf la solución de los problemas y el contenido de las preguntas frecuentes.	
Observaciones: Solo funciona para las soluciones y las preguntas frecuentes.	
Prototipo de interfaz:	

Tareas de Ingeniería

Tarea de Ingeniería	
Número Tarea: 9.1	Número Historia de Usuario: HU_9
Nombre Tarea: Exportar la solución del problema y el contenido de las preguntas frecuentes a formato pdf.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 11/02/2012	Fecha Fin: 13/02/2012
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita exportar la solución del problema y el contenido de las preguntas frecuentes a formato pdf.	

X. Imprimir.

Historia de Usuario	
Número: HU_10	Nombre Historia de Usuario: Imprimir.
Modificación de Historia de Usuario Número: ninguna	
Usuario: José Gustavo Suárez Matilla	Iteración Asignada: 4
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: Debe permitir imprimir la solución de los problemas y el contenido de las preguntas frecuentes.	
Observaciones: Solo funciona para las soluciones y las preguntas frecuentes.	
Prototipo de interfaz:	



Tareas de Ingeniería

Tarea de Ingeniería	
Número Tarea: 10.1	Número Historia de Usuario: HU_10
Nombre Tarea: Imprimir la solución del problema y el contenido de las preguntas frecuentes.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 11/02/2012	Fecha Fin: 13/02/2012
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita imprimir la solución del problema y el contenido de las preguntas frecuentes.	

XI. Mostrar la solución del problema.

Historia de Usuario	
Número: HU_11	Nombre Historia de Usuario: Mostrar la solución del problema.
Modificación de Historia de Usuario Número: ninguna	
Usuario: José Gustavo Suárez Matilla	Iteración Asignada: 4
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: Debe permitir mostrar la solución del problema identificado según la selección del usuario.	
Observaciones: Para mostrar la solución, primero se debe seleccionar la categoría, subcategoría y pregunta(s) que identifican el problema del que se mostrará la solución.	
Prototipo de interfaz:	



Tareas de Ingeniería

Tarea de Ingeniería	
Número Tarea: 11.1	Número Historia de Usuario: HU_11
Nombre Tarea: Mostrar solución.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 13/02/2012	Fecha Fin: 17/02/2012
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita mostrar la solución del problema.	

XII. Mostrar el contenido de las preguntas frecuentes.

Historia de Usuario	
Número: HU_12	Nombre Historia de Usuario: Mostrar el contenido de las preguntas frecuentes.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoency Leyva Ochoa	Iteración Asignada: 4
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: Debe permitir mostrar el contenido de las preguntas frecuentes seleccionadas por el usuario.	
Observaciones: Para mostrar el contenido primero se debe seleccionar la pregunta frecuente.	
Prototipo de interfaz:	



Tareas de Ingeniería

Tarea de Ingeniería	
Número Tarea: 12.1	Número Historia de Usuario: HU_12
Nombre Tarea: Mostrar solución.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 17/02/2012	Fecha Fin: 22/02/2012
Programador Responsable: Yoency Leyva Ochoa	
Descripción: Implementar la funcionalidad que permita mostrar el contenido de la pregunta frecuente seleccionada por el usuario.	

XIII. Mostrar categorías.

Historia de Usuario	
Número: HU_13	Nombre Historia de Usuario: Mostrar Categorías.
Modificación de Historia de Usuario Número: ninguna	
Usuario: José Gustavo Suárez Matilla	Iteración Asignada: 4
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: El objetivo de esta HU es mostrar (listar) las categorías que conforman el SA.	
Observaciones:	



Tareas de Ingeniería

Tarea de Ingeniería	
Número Tarea: 13.1	Número Historia de Usuario: HU_13
Nombre Tarea: Mostrar categorías.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22/02/2012	Fecha Fin: 29/02/2012
Programador Responsable: José Gustavo Suárez Matilla	
Descripción: Implementar la funcionalidad que permita mostrar las categorías del SA.	

Anexo 4 : Carta de aceptación del producto por parte del jefe del Departamento SIMAYS.



CARTA DE ACEPTACIÓN DEL CLIENTE

Yo, Jefe del Departamento SIMAYS certifico por este medio que el Trabajo de Diploma, titulado “**Sistema de Ayuda para la Edición Nova Escritorio**”, realizado en el **Departamento SIMAYS** cumple con los objetivos trazados inicialmente. Ha pasado satisfactoriamente por las pruebas y está listo para ser utilizado en la Edición Nova Escritorio 2013, como parte del proceso de migración de Plataformas de Código Abierto que experimenta Cuba como parte del proceso de informatización de la sociedad.

Y para que así conste, se firma la presente a los 23 días del mes de mayo de 2012.

A handwritten signature in blue ink, appearing to read 'Yoandy', is written over a horizontal line.

Ing. Yoandy Pérez Villazon
Jefe del Departamento SIMAYS

Universidad de las Ciencias Informáticas
Carretera a San Antonio Km 2 ½. Torrens. Boyeros. Ciudad de La Habana, Cuba.

Anexo 5 : Carta de aceptación del producto por parte del cliente del producto.



CARTA DE ACEPTACIÓN DEL CLIENTE

El cliente principal del Trabajo de Diploma, titulado **Sistema de Ayuda para la Edición Nova Escritorio** realizado en el **Departamento SIMAYS**, certifica por este medio su satisfacción con el mismo en correspondencia con los objetivos trazados.

Y para que así conste, se firma la presente a los 22 días del mes de mayo de 2012.

A handwritten signature in blue ink, appearing to read 'D. Bahr', is written above a horizontal line.

Firma del cliente
Ing. Daniel Hernandez Bahr
Jefe del Proyecto Nova Escritorio

Universidad de las Ciencias Informáticas
Carretera a San Antonio Km 2 ½. Torrens. Boyeros. Ciudad de La Habana. Cuba.

Glosario de siglas y términos

- **Asistencia activa:** Tipo de ayuda brindada por los Sistemas de Ayuda Inteligentes cuando detectan errores potenciales del usuario utilizando el sistema informático.
- **Asistencia pasiva:** Tipo de ayuda brindada por los Sistemas de Ayuda Inteligentes cuando el usuario de un sistema informático la solicite.
- **Ayuda de escritorio o soporte técnico (*help desk*):** Plantilla de soporte técnico que ayuda a solucionar los problemas de los usuarios relativos a *hardware* o *software* o consultan esos problemas con quien pueda resolverlos.
- **Ayuda emergente (*pop-up*):** Sistema de ayuda en línea cuyos mensajes aparecen como ventanas emergentes cuando el usuario pulsa sobre un tema o una determinada zona de la pantalla sobre el que se desea obtener ayuda.
- **Ayuda sensible al contexto (*context-sensitive help*):** Una forma de ayuda en la que un programa proporciona información de ayuda en pantalla al usuario que concierne al comando actual o a la operación que se está intentando llevar a cabo.
- **IA:** La Inteligencia Artificial (IA) es considerada una rama de la computación y relaciona un fenómeno natural con una analogía artificial a través de programas de computadora. La inteligencia artificial puede ser tomada como ciencia si se enfoca hacia la elaboración de programas basados en comparaciones con la eficiencia del hombre, contribuyendo a un mayor entendimiento del conocimiento humano. Si por otro lado es tomada como ingeniería, basada en una relación deseable de entrada-salida para sintetizar un programa de computadora. El resultado es un programa de alta eficiencia que funciona como una poderosa herramienta para quien la utiliza.
- **Regla:** Las reglas se utilizan en Prolog para significar que un hecho depende de uno o más hechos. Son la representación de las implicaciones lógicas del tipo $p \rightarrow q$ (p implica q). Consiste en una cabeza y un cuerpo, unidos por el signo ":-". La cabeza está formada por un único hecho el cuerpo puede ser uno o más hechos (conjunción de hechos), separados por una coma (","), que actúa como el "y" lógico. Estas finalizan con un punto (".").
- **Sistema Experto:** Es una rama de la IA que hace un uso del conocimiento especializado para resolver problemas como un especialista humano.
- **UML:** Es un lenguaje de modelado visual de propósito general, que es empleado para especificar, visualizar, construir y documentar los artefactos de un sistema de *software*.