

Universidad de las Ciencias Informáticas

Facultad 6



Título: Sistema basado en casos para la asistencia al proceso de soporte y ayuda en la gestión de incidencias tecnológicas en la Universidad de Ciencias Informáticas.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas.

Autores:

Orlando Cruz Rojas

Claudio Puig González

Tutores:

Ing. Janett Tase Hernández

Ing. Yunier Saborit Ramírez

Consultante:

Sergio Jesús García De La Puente

Ciudad de La Habana, junio de 2009

*"Si un hombre no tiene sus conocimientos en orden,
cuantos más posea mayor será su confusión."*

Herbert Spencer. Filósofo, psicólogo y sociólogo británico.

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 junio del año 2009.

Orlando Cruz Rojas

Firma del autor

Claudio Puig González

Firma del autor

Ing. Janett Tase Hernández

Firma de la tutora

Ing. Yunier Saborit Ramírez

Firma del tutor

DATOS DE CONTACTO

Nombre y Apellidos del **Tutor**: Yunier Saborit Ramírez

Dirección de Informatización

Teléfono: 837 2418

E-mail: yony@uci.cu

Currículum: Graduado de Ingeniero en Informática en el 2004. Realizó su primera actividad profesional en proyectos ese mismo año en el apoyo informático a la misión milagro. A partir de 2005 se incorpora a la Dirección de Informatización como especialista, cumpliendo el rol de arquitecto de la dirección. En el año 2006 pasa a ser Director de Informatización, cargo que ocupa actualmente. Ha tutorado varias tesis desde su graduación. Resumen de Tesis que ha tutorado: Sistema de Apoyo Logístico a la Misión Milagro(2004-2005), Análisis y Diseño del Catálogo de Servicios UDDI para la Informatización de la UCI (2006-2007), Propuesta de Arquitectura para la Migración a Software Libre del Sistema de Gestión Académica, Akademos (2006-2007). Sistema para la informatización del Convenio Cuba-Venezuela. Módulo ficha mixta (2006-2007).

Nombre y Apellidos de la **Tutora**: Janett Tase Hernández.

Dirección de Informatización

Teléfono: 837 2418

E-mail: jtase@uci.cu

Currículum: Graduada de Ingeniería Informática en el ISPJAE en el 2005. Después de graduada continuó su trabajo en la UCI, como Especialista de la Dirección de Informatización de este centro. Paralelamente se desempeñó como profesora de Algebra Lineal y Matemática Discreta impartida como parte del Plan de Estudio de los estudiantes de dicha Universidad, también ha sido tutora de estudiantes que optan por el título de Ingeniero en Ciencias Informático y parte de tribunal para el otorgamiento de este título. Su labor en la Dirección ha estado dirigida al control, gestión y seguimiento de proyectos de informatización de los procesos del centro.

Nombre y Apellidos del **Consultante**: Sergio Jesús García De La Puente.

E-mail: sgarcia@uci.cu

Graduado en el curso 2007-2008 de Ingeniero en Ciencias de la Informática, en la Universidad de las Ciencias Informáticas. Se ha desempeñado como profesor de pregrado en la asignatura de Ingeniería de Software con dos años de experiencia laboral. Ha cursado cursos de superación posgraduada en temáticas afines a su especialidad, así como en la contribución de su formación como docente.

Está vinculado al grupo de Migración y Soporte al Software Libre, adscrito al polo de Software Libre de la Facultad 10 de la Universidad de las Ciencias Informáticas. Se desempeña en los roles de Analista de sistema y como asesor en temas de Arquitectura de Software. Está vinculado a un grupo de desarrollo de software. Pertenece al grupo de Migración y se desempeña como especialista de Migración Parcial.

Ha presentado temas de investigación en los eventos UCIENCIA 2009 e Informática 2009.

Participó en el Comité Organizador del IV Taller Internacional de Software Libre y estándares abiertos de Software, celebrado en el marco de la XIII Convención y Feria Internacional Informática 2009, celebrada en La Habana, Cuba, durante los días del 9 al 13 de febrero de 2009.

Agradecimientos:

A mi madre, por traerme al mundo, por enseñarme a ser un luchador.

A mi abuela, por saber continuar mi educación y criarme como unos más de sus hijos.

A mi querida esposa, que me da su cariño cada día.

A mis suegros, que se ocupan de mí como padres.

A toda mi familia.

A Sergio, por ayudarnos y asesorarnos siempre que nos hizo falta.

A Abel Meneses, por su ayuda y sus consejos.

A Gladys, por la metodología.

Al profe Jorge Luis Vázquez, por sus revisiones.

A Karel y su novia, por su ayuda y su amistad.

A los tutores Janet y Yunier, por saber guiarnos hasta la meta.

A Keyla, por su hospitalidad y siempre colaborar con nosotros.

A todos mis amigos del GAT y de la UCI que siempre me preguntan como va la tesis.

A todos los profesores que tuvieron que ver con mi formación a lo largo de mi carrera y de mi vida escolar.

A Jorge Luis y Alberto, por sus buenas ideas y buenos consejos.

A nuestro tribunal, Trinchet, Aidacelys y Richard, por sus recomendaciones y su paciencia.

A mi gente de la puerta de servicio, que siempre me han brindado su amistad.

A Rita, que nos ayudo en el momento preciso.

A la Decana Yanet, que siempre se ha preocupado de nuestro CPT.

A mi amigo Luis Miguel, que me ha brindado siempre su confianza y amistad.

A Roberto, Henry y Alberto, mis amigos de telefonía.

A mi Brigada, por su apoyo.

A Pedro Piñeiro, por sus conocimientos y sus 2 buenos libros.

Un agradecimiento especial para mis amigos del Grupo de Desarrollo.

Muchas gracias y mis respetos para todos.

Claudio.

A mi mamá por estar en cada momento de mi vida siempre a mi lado, apoyándome y dando fuerzas para seguir adelante.

A mi papá, porque además de ser mi padre ha sido más que un amigo.

A mi hermana por estar justo a mi lado cuando la he necesitado.

A mi esposa por ayudarme en la superación tanto profesional como personal. Por su sinceridad por encima de cualquier resultado.

A mi hermano Henry por darme la mano sincera.

A mis tías, tíos, primas, primos, familia en general por estar siempre atentos a mi y a mis estudios.

A mi familia tunera que siempre me acogió en su seno como un hijo más. Gracias Maritza, Fernando, Fernandito y Reynier.

Al Grupo de Desarrollo de la DGT, Carlos, Lázaro, Halena y El "Norbert", por apoyarme incondicionalmente en este proyecto.

A Aidacelys, Trinchet y Pompa por ser consecuentes y exigentes como tribunal.

Para Aidacelys un reconocimiento especial.

A Rudel y Keila por ser más que profesores, amigos.

A Michael por darme la oportunidad de trabajar junto a él.

A Sergito, Gladys y Abel por su asesoramiento.

A Janett y Yunier por la tutoría de este trabajo. Su guía resultó importante en las experiencias adquiridas durante el desarrollo de la tesis.

A Jorge Luis el de PSCT por ayudarnos a entender los grandes problemas de la ciencia y la tecnología.

A Jorge Luis mi hermano santiaguero por ser mi hermano internacionalista.

A Alberto mi hermano granmense.

A todos los integrantes de la DIN, sin excepción.

A todos mis compañeros de trabajo, que de una forma u otra han cooperador en mis estudios y en el desarrollo de este trabajo de tesis.

A todos mis amigos y compañeros gastronómicos por ayudarme con la merienda escolar.

A Laritza, Olguita y Lucmary por su guía profesional.

A Pedro Piñeiro por su ayuda en la investigación y desarrollo de este trabajo de tesis.

A René, Fito, La chiqui y todos los compañeros de alimentos por su almuerzo sabatino.

A Karell y Jose por ser amigos míos.

A mis compañeros de INDER (Marielena, Melvyn, todos)

A todos los médicos por ejercer la profesión más importante al salvar vidas humanas.

A todos mis compañeros de ómnibus donde viajamos a diario y formar parte de mi familia.

A todos los profesores que me han permitido obtener los conocimientos necesarios para realizar este trabajo.

A la Universidad de las Ciencias Informáticas y a la Revolución, por haberme dado la posibilidad de realizar mis estudios.

Y a todos los que no puse, pero que de una forma u otra han contribuido a mi formación profesional y personal.

A todos, gracias.

Landy

Dedicatoria:

A mi madre, que le debo todo lo que soy y he alcanzado en mi vida.

A mi abuela, mi segunda madre, que siempre ha estado ahí para apoyarme y darme su cariño.

A mi padre, que a pesar de la distancia siempre piensa en mí.

A mi esposa, por ganarse mi cariño, darme su confianza y apoyo.

Claudio.

A esa familia grandiosa que tengo.

A mis padres que les debía este alegrón.

A Isis mi sobrina, que hace tiempo no la beso.

A Kenia mi esposa y amiga, este periodo ha sido duro.

A todos, todos mis amigos y enemigos, por servirme de motor impulsor.

Especial para los amigos que ya no están conmigo para celebrar.

Landy.

Resumen:

La Universidad de Ciencias Informáticas (UCI) cuenta con una amplia y diversa infraestructura tecnológica, la cual sufre roturas y deterioro que afectan los procesos que en ella se llevan a cabo. Actualmente en el proceso de solución de estas roturas no existe un mecanismo donde quede constancia de las experiencias adquiridas por el personal a cargo del soporte técnico.

El objetivo fundamental que persigue esta investigación es el diseño e implementación de un sistema informático basado en casos, que sirva de herramienta al proceso de soporte o ayuda en la gestión de incidentes tecnológicos, y donde quede plasmado el conocimiento referente a la solución de incidentes tecnológicos, de los técnicos y especialistas encargados de la asistencia técnica.

Dentro de sus principales funciones cuenta con un módulo de almacenamiento, una base de casos, donde se almacenará la experticia en el dominio de las tecnologías en forma de casos. Los casos no son más que experiencias adquiridas por los expertos en situaciones previas, y que permitirán dar soluciones en dependencia al grado de semejanza con nuevos casos que ocurran, en este proceso queda evidenciado otras funcionalidades, que incluyen la recuperación de casos semejantes y adaptación a partir de un caso dado.

Esta puede ser implementada en centros de soportes con el fin de brindar una primera línea de ayuda o ayuda primaria en la gestión de incidentes tecnológicos, siendo estos centros único punto de contacto para la gestión de servicios de tecnología informática según ITIL.

Palabras claves: infraestructura tecnológica, soporte técnico, sistema informático basado en casos, incidentes tecnológicos, base de casos, experticia, dominio de las tecnologías, centros de soportes, primera línea de ayuda

Tabla de Contenidos

Agradecimientos	I
Dedicatoria.....	V
Resumen	VI
Introducción	1
Capítulo 1	7
Fundamentación Teórica.	7
1.1 Gestión del conocimiento.....	7
1.2 Biblioteca de Infraestructura de Tecnologías de Información (ITIL).....	9
1.3 SBC	11
1.4 Sistemas Basados en Casos (SBCs)	15
1.4.1 Aprendizaje automatizado (AA).....	17
1.4.2 Módulos de los SBCs.....	18
1.5 SBC existentes	24
1.6 Tecnología, técnicas y herramientas.....	25
1.6.1 Uso de Metodologías Ágiles:.....	26
Capítulo 2	29
Descripción del Sistema.....	29
2.1 Objetivos estratégicos.....	29
2.2 Flujo actual de proceso	29
2.3 Objeto de automatización.....	29
2.4 Modelo de Dominio	30
2.5 Lista de Reserva del Producto (LRP)	31
2.5.1 Funcionalidades del sistema	31
2.5.2 Requisitos no funcionales del sistema.....	34
2.6 Plan de Releases	37
2.7 Historias de usuarios.....	38
2.8 Descripción de la Arquitectura	58
2.8.1 Modelo Vista Controlador.....	58
2.8.2 Diseño con metáforas	59
2.9 Modelo de datos	61
Capítulo 3	64

Desarrollo del Sistema.....	64
3.1 Plantilla código fuente.....	64
3.2 Estándar de código.....	69
3.3 Validación de la solución propuesta.....	70
3.4 Acerca de las funcionalidades obtenidas.....	72
3.5 Conclusiones.....	73
Conclusiones Generales.....	74
Recomendaciones.....	75
Referencias bibliográficas.....	76
Anexos	
Anexo 1. Tipos de Sistemas Basado en el Conocimiento	
Anexo 2. Guión de la Metodología SXP	
Anexo 3. Plantilla Historia de Usuario	
Anexo 4. Plantilla Caso de Prueba de Aceptación	
Glosario de Términos.....	80

Introducción

Desde la antigüedad el hombre ha buscado soluciones a cada uno de los problemas que se han presentado con el objetivo de mejorar su forma de vida en todos los aspectos. Estas soluciones han sido transmitidas por generaciones, evolucionando y convirtiéndose en patrones a seguir para enfrentar cada situación que se presenta.

Los avances en las tecnologías informáticas han propiciado un incremento de las capacidades de adquirir, almacenar y transmitir grandes cantidades de información, con una gran disminución de costos; no obstante se dificulta la habilidad para gestionar y utilizar correctamente las masas de información y conocimientos disponibles correctamente.

Se genera entonces el desarrollo de herramientas dentro de la industria del software para gestionar el conocimiento, con el objetivo de proporcionar un elemento de apoyo que permita, sobre la base del manejo del conocimiento de las organizaciones, tomar mejores decisiones.

Las herramientas tecnológicas para la administración o gestión del conocimiento, como otras, están diseñadas para facilitar el trabajo y permitir que los recursos sean utilizados eficientemente, intercambiando información y conocimiento dentro y fuera de dichas organizaciones.

En la actualidad existe gran variedad de productos de software que pueden ayudar a las organizaciones a alcanzar tales objetivos. En dependencia a las características y/o necesidades de cada organización, estas requerirán de distintos tipos de herramientas. Para algunas organizaciones la clave de la gestión del conocimiento se encuentra en adquirir ventaja de sus propios empleados y ayudar a encontrar experiencias dentro de ellas. Otras enfocan su estrategia en la optimización y el mejor aprovechamiento de las prácticas disponibles para tomar decisiones más rápidas.

Paradójicamente, uno de los problemas que puede surgir al hacer difusión del conocimiento es generar más del que la organización tiene capacidad de manejar. No se trata sólo de codificarlo y tener varios sistemas que generen información, sino que tener una estrategia de gestión del conocimiento es vital, pues las herramientas no resolverán ningún problema por sí solas.

El éxito en la adquisición o desarrollo de una herramienta tecnológica, depende mayormente de factores no técnicos como son: el factor humano, los procesos organizacionales y la cultura en general.

En Cuba, en las décadas del '60 y el '70, se creó un plan de desarrollo en la rama de las ciencias computacionales para la preparación de especialistas cubanos en países desarrollados con el propósito de adiestrar a otros en nuestro país, a partir de los conocimientos adquiridos.

Especialistas de La Universidad de la Habana y la ISPJAE fueron exponentes de este plan, lo que posibilitó el desarrollo de las primeras herramientas o productos de software en Cuba. Desde su creación en el año 2002, con el objetivo de formar profesionales en esta rama y contribuir al desarrollo de la industria informática cubana, la UCI ha creado diversas herramientas tecnológicas, desarrollando entre otras las relacionadas con la gestión de la información y el conocimiento.

La UCI cuenta con una amplia infraestructura tecnológica, entre equipos y periféricos, los cuales sufren una explotación diaria y se ven afectados por roturas o desperfectos. Dentro de las principales causas de esas roturas se pueden mencionar:

- La explotación de los equipos en condiciones climáticas inadecuadas.
- El poco conocimiento de los usuarios sobre el uso de las tecnologías.
- El mal uso y maltrato del equipamiento.
- El no cumplimiento de los protocolos de seguridad informática.

Para solucionar estos problemas en la infraestructura existe la Dirección de Gestión Tecnológica (DGT), que está conformada por cuatro grupos especializados: Asistencia Técnica (GAT), Redes, Televisión y Telefonía, estos son los encargados de todo las ramas del soporte técnico. Actualmente esta dirección cuenta con mayor número de técnicos con relación a la cantidad de especialistas, estos últimos diestros en materia de tecnología.

El GAT ofrece un servicio de 24 horas para la gestión de problemas tecnológicos, a través del número telefónico 6060, sirviendo como único punto de contacto con los usuarios de la comunidad UCI. También se utiliza GatServer, herramienta que gestiona el proceso de los problemas tecnológicos y que tiene como alcance el área de servicios técnicos, es decir, sólo es utilizado por el personal de la DGT.

Este proceso de gestión de incidentes se inicia por la vía antes mencionada, interactuando el usuario con la recepcionista de llamadas. Esta trabaja directamente con GatServer, y mediante un formulario recoge toda la información referente al usuario afectado (número de solapín, usuario del dominio,

teléfono, edificio, apto o local, horario en el que se puede realizar la visita), así como los datos del equipo roto y descripción asociada (tipo de equipo, descripción del usuario).

Al generarse el reporte éste es asignado por GatServer, automáticamente, al técnico que atiende el área donde fue ubicado. El técnico diagnostica la rotura o el mal funcionamiento del equipo reportado, si determina que el problema es de configuración o reinstalación de algún software, procede a resolverlo. Si el resultado es satisfactorio el técnico comunica al Jefe de brigada que ha concluido el trabajo, quien a su vez efectúa el cierre del reporte, de lo contrario el reporte es asignado al grupo especializado que atiende ese equipo.

Los especialistas de los grupos realizan un análisis más profundo de la situación, en dependencia del nivel de rotura que exista estos intentan solucionar el problema. Si el problema es solucionado, se realiza el mismo procedimiento con el jefe de brigada para el cierre del reporte. Si no es solucionado entonces el reporte es asignado a los especialistas de Copextel: entidad que ofrece el servicio de reparación de hardware de tecnología.

En esta parte del proceso debe solucionarse el problema, ya que dentro de los servicios que esta entidad ofrece está el cambio de piezas y partes de hardware; en el caso que no exista la solución, se cambia el equipo y se declara baja técnica.

Con el objetivo de mejorar este proceso se realizó un estudio acerca de los problemas tecnológicos que suceden a diario en la Universidad. Para ello se analizaron dos meses con características diferentes, septiembre y noviembre, con el propósito de medir el comportamiento de los reportes registrados en diferentes períodos. Se elige septiembre porque al inicio del curso escolar los equipos sufren un cambio brusco, ya que durante el período Julio – Agosto no se encuentran en uso, y en ocasiones son víctimas de la humedad ambiental. También se elige noviembre que, por el contrario, presenta un máximo uso de las tecnologías.

En el mes de septiembre se reportó un gran número de problemas tecnológicos, registrándose un promedio de cien reportes diarios. Del total de reportes registrados en ese período fueron solucionados un 50 % por los técnicos y especialistas de la DGT.

En el mes de noviembre se registró un menor número de problemas tecnológicos con respecto a septiembre, para un promedio de setenta reportes diarios. Aun así, del total de reportes registrados en ese período solo fueron solucionados un 60 % por el personal de la DGT.

Gran número de los reportes que se realizan se deben a problemas de configuración de software, los cuales pueden ser solucionados sin la visita de un técnico o especialista. En muchas ocasiones estos problemas pueden ser resueltos por el propio usuario en el inicio de este proceso, si se ofreciera una primera línea de soporte (atención telefónica especializada) que permita guiarlo en la solución del problema.

El proceso de gestión de servicios de TI (Tecnología de la Información), que implementa la DGT, aún no satisface las expectativas de los usuarios, ya que existe una inconsistente calidad de respuesta a los llamados, demora para solucionar los problemas, así como baja confianza y/o conocimiento del cliente.

El sistema que utiliza la DGT para gestionar los problemas tecnológicos a partir de una segunda línea de ayuda presenta limitaciones. Una limitante es que no permite comparar nuevos problemas con otros registrados anteriormente, lo que trae como consecuencia una innecesaria asignación de éstos a la segunda línea de soporte. De esta forma, gran número de los problemas que pudieran ser solucionados por los propios usuarios en la primera línea de soporte sobrecargan el sistema, y con ello el trabajo de los técnicos, dificultando el tiempo de respuesta a los reportes.

A diario aparecen problemas que son resueltos de forma reactiva, sin embargo, estos se repiten frecuentemente de forma similar, mas no se enfoca el trabajo proactivo para así identificar y resolver problemas y errores conocidos antes de que sucedan. Esto conlleva a tener que resolver los mismos problemas repetidamente en vez de prevenirlos, pues las soluciones podrían ser almacenadas para uso futuro.

El desconocimiento acerca de las categorías de problemas tecnológicos conlleva a una incorrecta escalabilidad en la gestión de los mismos. El servicio de respuesta carece de un mecanismo que gestione el conocimiento adquirido en la resolución de los problemas y que permita comparar y compartir las experiencias de los especialistas y técnicos. Por esta causa, existe gran dependencia del personal especializado, que generalmente soluciona los reportes. Por consiguiente no es posible

brindar respuestas en menor plazo de tiempo, no repetir errores y hacer uso de las mejores prácticas, según el nivel de Gestión de Servicios que merecen los usuarios de la comunidad UCI.

A partir de la situación antes descrita se define como **problema científico**:

¿Cómo contribuir al mejoramiento del proceso de soporte o ayuda para la gestión de incidentes tecnológicos en la UCI?

Objeto de estudio: Los Sistemas Basados en el Conocimiento (SBC).

Campo de acción: Los Sistemas Basados en Casos (SBCs) para la gestión de incidencias tecnológicas en la UCI.

Objetivo general: Desarrollar un sistema informático basado en casos que contribuya al mejoramiento en la gestión de incidentes tecnológicos en la UCI según la metodología ITIL.

Para cumplir el objetivo general han sido trazados los siguientes **objetivos específicos**:

- Definir las funcionalidades que tendrá el sistema.
- Diseñar el sistema.
- Implementar y probar el sistema.

Tareas científicas a desarrollar para cumplir los objetivos específicos:

- Investigación sobre los SBC y los SBCs empleando Razonamiento Basado en Casos (RBC) para la creación de bases de conocimientos, además de su utilización en sistemas de auto-gestión del conocimiento.
- Estudio de los procesos de soporte de tecnologías que actualmente funcionan en la UCI.
- Estudio de las líneas bases de la arquitectura para el desarrollo de sistemas informáticos en la UCI propuesta por la Dirección de Informatización.
- Realización de las listas de reservas de productos.
- Realización del diseño de la aplicación usando la metodología de desarrollo seleccionada.
- Implementación de la Base de Casos (BCs).
- Implementación del Motor de Inferencia (MI).
- Implementación de la Interfaz de Usuario.

Estructura de tesis

El trabajo de tesis está estructurado en tres capítulos, en el primero de ellos se estudian los conceptos y características relacionadas con los SBC, específicamente enfocadas a los Sistemas Basados en Casos. Se realiza un estudio del razonamiento basado en casos, rama de la Inteligencia Artificial (IA), e ITIL como marco de trabajo para la gestión de servicios en los centros de soporte.

El segundo capítulo contiene un marco conceptual asociado a la información que será manipulada por el sistema, obteniéndose las funcionalidades, requisitos deseados y el objeto de automatización.

El tercer capítulo se centra en la implementación y la construcción de la estructura de la aplicación y se presenta la solución propuesta.

Capítulo 1

Fundamentación Teórica.

En este capítulo se realiza un estudio acerca de herramientas o sistemas utilizados para la gestión del conocimiento y técnicas de razonamiento. Son analizadas novedosas metodologías para la gestión de incidentes tecnológicos, el proceso de gestión y desarrollo de software.

1.1 Gestión del conocimiento

“Gestión del Conocimiento es, de manera resumida, la obtención del conocimiento necesario por las personas adecuadas, en el tiempo, forma y lugar adecuados.”[1]

“Es un proceso sistemático e intencionado de creación, compartición y aplicación de conocimiento crítico para el desarrollo de la estrategia de negocio, las decisiones u operaciones que conlleva.”[2]

La gestión del conocimiento en las empresas exige la constitución de políticas claras dirigidas a la valoración de activos intangibles y a la administración del conocimiento. Es necesario tener presente que las personas son las poseedoras de los conocimientos y experiencias que agregan valor a la empresa. [3]

Al gestionar conocimiento se realiza conservación, uso y comunicación del mismo; se aprovecha la experiencia del personal de la empresa para implementar sistemas informáticos en un contexto de trabajo con el objetivo de optimizar la labor cotidiana.

El conocimiento tiene algunas características propias, tales como:

- Representa un activo intangible.
- Difícil de medir.
- Volátil, puesto que evoluciona.
- Inagotable, pues se desarrolla con su uso.
- Se puede copiar y reproducir sin límites.
- Se puede transferir sin pérdida.
- Se utiliza simultáneamente en diferentes procesos.
- Se desarrolla mediante el aprendizaje.
- Justifica el uso de una metodología especializada donde se involucran las personas, la cultura, los procesos y la tecnología.

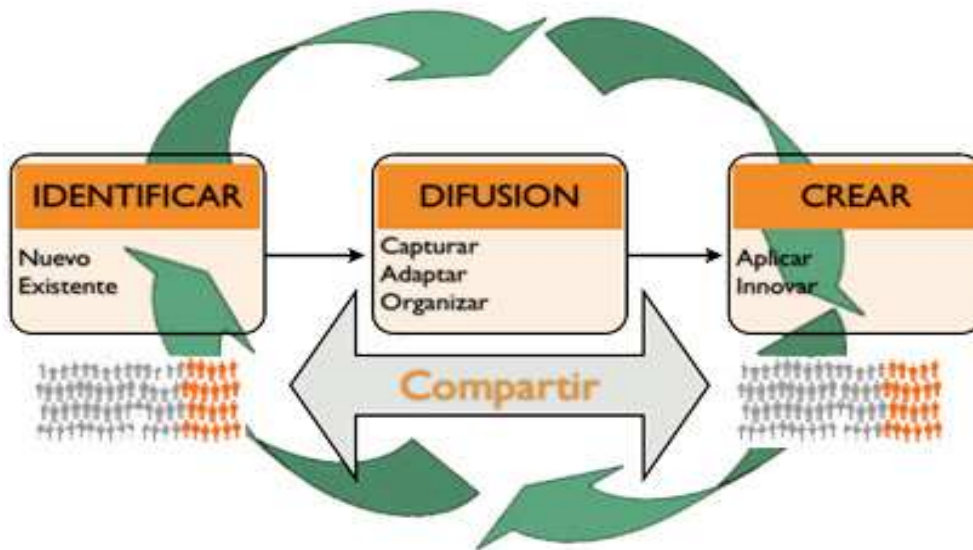


Fig. 1 Proceso de gestión del conocimiento.

Cuando no se gestiona el conocimiento la experiencia se pierde en medio de la rutina laboral, debido a la memoria de las personas, o desaparece cuando las personas dejan de trabajar en una empresa.

Aquí radica la importancia de almacenar el conocimiento que surge de la experiencia ante diversas situaciones.

La clave está en contar con los dispositivos o herramientas adecuadas que permitan almacenar con seguridad el conocimiento, además de fácil acceso para las personas autorizadas en una organización que utilice esta experiencia en su labor diaria.

El conocimiento puede ser almacenado por diferentes vías y representado en diferentes formatos, tales como: documentos, manuales, prácticas o políticas organizacionales; además pueden utilizarse dispositivos electrónicos para crear bases de datos o prácticas codificadas que reúnan las lecciones aprendidas y las mejores prácticas en una empresa.

La gestión del conocimiento ayuda a las empresas a incrementar la calidad, a reducir los tiempos de respuesta y a mejorar la operación al facilitar la solución de problemas y la implementación de soluciones adecuadas en base a las experiencias obtenidas en el pasado. Igualmente establece metodologías, sistemas y la contribución de personas para almacenar el conocimiento que surge de la experiencia diaria en las organizaciones: cómo solucionar situaciones recurrentes, qué hacer con los incidentes frecuentes, adaptar mecanismos ya conocidos para corregir nuevos sucesos, entre otros.

La gestión del conocimiento restaura con rapidez la normalidad luego de un incidente, esto se aplica en las áreas de TI para brindar un servicio continuo y de calidad en las empresas.

1.2 Biblioteca de Infraestructura de Tecnologías de Información (ITIL)

De acuerdo con ITIL, se busca implementar las mejores prácticas de TI para incrementar la calidad de los servicios, y lograr una exitosa gestión de incidentes, para lo que se hace indispensable la utilización de bases de conocimientos.

Pueden hallarse bases de datos con errores conocidos que describan cómo reconocer los incidentes, soluciones y workarounds (soluciones temporales disponibles).

La gestión de problemas descubre la causa raíz de los incidentes en la infraestructura de TI para minimizar el impacto desfavorable que producen. Por tanto, permiten iniciar acciones para mejorar o

corregir la situación. Se puede relacionar fácilmente los problemas con los incidentes y los cambios, y así rápidamente los problemas con su solución.

La gestión de problemas en las TI con éxito depende, según ITIL, de la eficiencia en el registro automatizado de incidentes y del comportamiento de la infraestructura, en donde las bases de datos juegan un papel fundamental para el proceso de solución de problemas.

La gestión de problemas consiste en buscar, descubrir y eliminar las causas de los mismos a fin de prevenir los incidentes, documentando una solución temporal que convierta al problema en un error conocido.

La aplicación de herramientas para gestionar el conocimiento genera importantes beneficios:

- Solución de problemas en menor tiempo.
- Llamadas más cortas que generan una reducción del gasto telefónico.
- Menores escalamientos a las áreas de segundo nivel.
- Mejores productos y servicios.
- Incrementa la satisfacción de los clientes.

Otras ventajas para las organizaciones:

- Aseguramiento de la memoria organizacional.
- Aseguramiento del flujo de conocimiento.
- Se facilita el trabajo colaborativo.

Aunque el uso de herramientas del conocimiento ayuda a elevar la satisfacción de los clientes, en el ámbito mundial no son aprovechadas al máximo.

De acuerdo con un estudio en los últimos años que produjo *ContactCenterWorld.com* denominado *International Contact Center Benchmarking Study (2005)*, en el que participaron 385 de los más grandes Centros de Contacto del mundo, se encontró que sólo el 28.5% utilizan la Gestión del Conocimiento para diversos fines, mientras que un 19.48% declararon que planean utilizarla en el futuro. [4]

Existen metodologías estándares para la gestión del conocimiento y que se utilizan para el desarrollo de SBC. Estas ofrecen teorías, métodos y técnicas científicas para la representación del conocimiento y proporcionan reglas para obtener los procesos de razonamiento usados por trabajadores del conocimiento, y reproducirlos en sistemas informáticos. Entre estas se encuentran: VITAL, MIKE, PROTÉGÉ-II, KADS y CommonKADS.

1.3 SBC

La inteligencia artificial (IA) se enfocó inicialmente en la producción de sistemas con capacidad de asistir al ser humano en la toma de decisiones o en la búsqueda de soluciones a problemas. Se orientó el desarrollo de mecanismos que facilitaran la comunicación con el computador en lenguaje natural. Se estructuraron sistemas expertos, actualmente conocidos como SBC, para almacenar la experticia humana. [5]

El conocimiento representado en los SBC es el de los expertos en el dominio, y una parte de éste consiste en la relación causa-efecto. Esta relación se origina a partir de experiencias pasadas y son llamadas heurísticas. Estas experiencias representan conocimiento informal o atajos que permiten al experto encontrar rápidamente solución a un problema, sin tener que realizar un análisis detallado de la situación, debido a la experiencia acumulada de casos resueltos anteriormente o intentos fallidos en resolver un problema similar. Éstos pueden recordar o no en detalles un análisis realizado a una situación anterior, pero pueden reconocer el enfoque dado a esta situación.

Las funciones heurísticas representan conocimiento general usado para guiar la búsqueda, mientras que la heurística en los SBC consiste en el conocimiento heurístico usado por un experto en la solución de un problema, esta es más específica del dominio y más profunda en naturaleza.

Un experto es una persona que posee destrezas que le permiten sacar conclusiones de experiencias pasadas y rápidamente enfocar un análisis del problema dado. Las posibilidades de éxito de un experto en la solución de problemas se deben a que ha adquirido un conjunto de relaciones de causa y efecto poderosas basadas en la experiencia. Éstos son capaces de manejar el conocimiento básico para reconocer rápidamente rasgos sobresalientes del problema, clasificar este de acuerdo a estas características y buscar una solución. Últimamente, se reconoce que en lugar de relaciones de causas

y efectos los expertos recuerdan casos similares al problema a resolver y toman de ellos las soluciones dadas como punto de partida para encontrar la solución al nuevo problema.

El experto se caracteriza por tener mayor amplitud en la variedad de problemas que puede resolver con respecto a otros profesionales del dominio, además de tener más rapidez y calidad en encontrar una solución.

El conocimiento en cualquier especialidad es usualmente de dos tipos: público y privado. El conocimiento público incluye las definiciones, hechos y teorías publicadas. Pero la experticia usualmente incluye más que esta clase de conocimiento. Los expertos humanos generalmente poseen conocimiento privado.

Esclarecer y reproducir tal conocimiento es la tarea central en la construcción de SBC; el sujeto de esta acción es el ingeniero de conocimientos. Sobre el conocimiento público hay consenso, el privado puede llevar a controversias entre los expertos.

Entre las razones que sustentan enfatizar en el conocimiento en lugar de en los métodos de razonamiento formal están:

- La mayoría de los problemas difíciles e interesantes no tienen soluciones algorítmicas tratables.
- Los expertos humanos obtienen su destacado desempeño gracias al conocimiento que poseen.
- El conocimiento es una fuente escasa cuyo refinamiento y reproducción crea riqueza.

En resumen, la potencia de un SBC radica en el conocimiento que posee.

Existen diferentes términos para referirse a éstos sistemas, tales como: sistemas basados en el conocimiento, sistemas expertos, sistemas expertos en el conocimiento.

Un SBC puede definirse como un sistema computarizado que utiliza conocimiento específico de un dominio y se emplea para dar solución a problemas en dicho dominio. Estos sistemas son un modelo computacional, de mayor nivel que el paradigma de la programación convencional y están formados por tres componentes: la base de conocimiento (BC), el MI y la interfaz. [6]

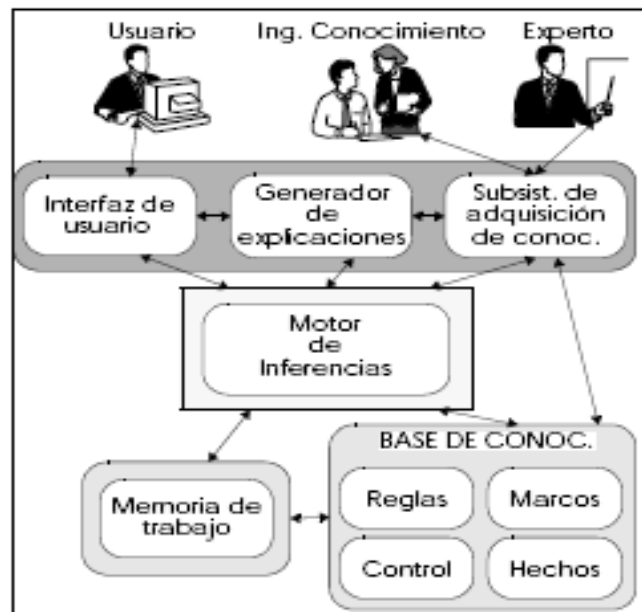


Fig. 2 Arquitectura de los Sistemas basados en el conocimiento.

En la BC se almacena el conocimiento necesario para resolver los problemas del dominio para el cual se desarrolla el SBC.

El MI es un procedimiento en el cual está implementado algún método que utiliza el conocimiento almacenado en la BC para resolver problemas del dominio. El tipo de conocimiento determina que método de solución de problemas (MSP) es posible utilizar.

Desarrollar un SBC comprende la realización de cada uno de sus componentes. La BC es el núcleo de estos sistemas y es particular para cada uno de ellos, por lo tanto es necesario desarrollarla para cada sistema. El MI y la interfaz de usuario pueden ser reutilizadas para el desarrollo de varios sistemas, siempre que éstos utilicen el mismo tipo de conocimiento y el mismo tipo de razonamiento.

El conocimiento almacenado en la BC puede ser de diferentes tipos: simbólicos, pesos de una red neuronal, ejemplos o casos de problemas del dominio, entre otros. Estos dan lugar a diferentes tipos de Sistemas Basado en el Conocimiento, entre los que se encuentran: Sistemas Basados en Reglas (SBR), Sistemas Basados en Frames, Sistemas Basados en Probabilidades y Sistemas Basados en Casos, entre otros. **Ver Anexo 1.**

- Sistemas Basados en Reglas.

Los SBR son SBC en los que la forma de representación del conocimiento usado son las reglas de producción y como método de inferencia utiliza la regla de modus ponens.

Los SBR son llamados frecuentemente sistemas de producción, una idea de 1943 expresada por E. L. Post. [7]

- Sistemas Basados en Probabilidades.

Los sistemas probabilísticos se consideran otro de los tipos más importantes de SBC. Sus partes fundamentales son la BC (con estructura de espacio probabilístico) y el MI (basado en las probabilidades condicionales). [8]

- Sistemas Basados en Casos.

Los Sistemas Basados en Caso utilizan el RBC, es decir, razonan en base a experiencias o "casos" previos. El RBC es una alternativa entre otras metodologías para construir SBC. Al razonar basado en casos el solucionador de problemas recuerda situaciones previas similares a la actual y las usa para ayudar a resolver el nuevo problema. [9]

Ventajas que proporcionan los SBC:

- Amplia distribución de experticia escasa.
- Fácil modificación.
- Consistencia en las respuestas.
- Gran accesibilidad.
- Preservación de la experticia (no "se pierde").
- Solución de problemas que incluyen datos incompletos.
- Explicación de soluciones.
- Permite evaluar el efecto de nuevas estrategias añadiendo o modificando conocimiento.
- Constituye un entrenador en el dominio de aplicación.

La estrechez del dominio, la complejidad y naturaleza del problema y de los expertos, las soluciones convencionales, material escrito entre muchas otras, son situaciones que propician la creación de los SBC. Estos se fundamentan en el uso de experiencia previa contenida en dos tipos de fuentes de

información principales: expertos (fuente primaria) y libros (fuente secundaria), formalizando este conocimiento mediante la construcción de casos.

Es por ello que se establecen las características y procedimientos a seguir para encontrar una solución en cada evento particular. Por otra parte, es importante establecer semejanza entre los eventos para de esta manera aprovechar las soluciones e inferencias encontradas previamente y mediante una cantidad pequeña de modificaciones poder estructurar y resolver otro caso que presente características similares. [10]

1.4 Sistemas Basados en Casos (SBCs)

En los últimos tiempos ha tomado auge a nivel internacional, dentro del campo de la IA, el RBC. Esta tecnología se utiliza para el desarrollo de SBC y da lugar a los SBCs. Esta técnica ha sido aplicada con éxito en diversos campos como: Ecología, Diseño, Música, Planificación, Medicina, Aviación, Leyes, entre otras. [11]

Un SBCs tiene dos componentes principales: una BCs y un solucionador de problemas. La BCs contiene las descripciones de los problemas resueltos o no previamente. El solucionador de problemas tiene a su vez dos componentes: un recuperador de casos y un razonador sobre el problema.

Para construir un SBCs deben tenerse en cuenta los siguientes elementos: ¿Cuál es el contenido de un caso? ¿Cuáles son los algoritmos de acceso? ¿Cuáles son los índices a los casos y los métodos de adaptación? ¿Qué tipo de casos se necesitan y cómo se escogen? ¿Cómo se escogen sus índices?

Los SBCs parten de problemas resueltos en un dominio de aplicación y mediante un proceso de adaptación encuentran la solución a un nuevo problema. Los sistemas que utilizan éste tipo de razonamiento pueden ser de dos tipos: interpretativos o solucionador de problemas. En el tipo solucionador de problemas el proceso que se ejecuta es "recordar un caso y adaptar su solución".

En los sistemas interpretativos, las situaciones nuevas se evalúan en el contexto de las soluciones viejas.

En los sistemas solucionadores de problemas, el estilo de solución se caracteriza por un fuerte componente del sistema para realizar el proceso de adaptación. Mediante este proceso se construye la

solución del nuevo problema a partir de la solución dada a una situación semejante. Este es el modelo de RBC más empleado para resolver tareas de diseño y planificación.

Un sistema con RBC se considera un solucionador de problemas si puede aprender de los errores (no repite el mismo error dos veces), si trata de encontrar una solución mejor aún cuando ya esté disponible una y si no considerara utilizable los resultados inmediatamente, primero los modificará para adecuarlos a las necesidades actuales.

El RBC es particularmente útil para resolver problemas y tomar decisiones cuando:

- Es difícil formular reglas pero existen casos disponibles.
- Las reglas pueden ser formuladas pero requieren mayor información de la que normalmente se dispone.
- Las reglas se pueden formular usarlas es costoso porque:
 - El número de reglas es muy grande.
 - La longitud promedio de la cadena de reglas necesaria para realizar una inferencia es muy larga.
- Casos con soluciones similares tienen una formalización del problema similar.
- Verificar la utilidad de la solución es fácil.
- Existen casos disponibles.

Cuatro aspectos importantes a tener en cuenta para lograr calidad y fiabilidad en los resultados producidos por estos sistemas son: la experiencia del sistema, su habilidad para comprender nuevas situaciones en términos de la experiencia, su habilidad en la adaptación y en la evaluación.

1- La experiencia del sistema.

Un sistema con menos casos almacenados en su BC tendrá menor experiencia para trabajar, pero no necesariamente sus respuestas serán peores que las dadas por un sistema de mayor experiencia, si es lo suficientemente creativo en la comprensión que hace del nuevo problema y en el proceso de adaptación. Una menor cantidad de casos puede sólo llevar a poder resolver menor diversidad de problemas, pero no afecta la calidad de las soluciones halladas.

2- Su habilidad para comprender nuevas situaciones en términos de la experiencia.

Esta habilidad se manifiesta en el proceso de recuperar los casos antiguos semejantes al nuevo problema, lo cual es el núcleo del sistema. El otro factor que influye es la interpretación que se realiza, o sea, el proceso de comparar la nueva situación con las experiencias recuperadas.

3- Su habilidad en la adaptación.

Las respuestas creativas resultan de aplicar las estrategias de adaptación de una manera creativa. Algoritmos de adaptación potentes pueden permitir obtener buenas soluciones a los nuevos problemas a partir de casos "no muy semejantes a los problemas".

4- Su habilidad en la evaluación.

Para aprender de su experiencia el sistema debe retroalimentarse de modo que pueda conocer lo que fue correcto y lo que no. De allí que la evaluación de la solución dada, y consecuentemente la auto-reparación, son contribuyentes importantes en la experticia del sistema.

En un sistema con RBC, el conocimiento es representado principalmente en forma de casos que describan experiencias concretas. En tanto, si fuera necesario, también otros tipos de conocimiento sobre el dominio de aplicación pueden ser almacenados, por ejemplo, casos abstractos y generalizados, tipos de datos, modelos de objetos usados como información. [12]

1.4.1 Aprendizaje automatizado (AA)

Otro aspecto relevante en los SBCs es cómo se manifiesta el aprendizaje, de ahí que el AA denote la posibilidad de cambiar que tiene el sistema, o sea, de ajustarse, de modo que permite al sistema hacer la misma tarea más eficiente la próxima vez, por lo que se puede ver como un proceso de reconstrucción del conocimiento del sistema.

El aprendizaje se manifiesta cuando se alcanza soluciones más precisas, cuando se cubre un rango mayor de problemas que pueden ser resueltos por el sistema y obtienen respuestas más exactas, simplificando el conocimiento almacenado, también cuando el sistema sea capaz de reordenar su base de conocimientos para que sea más legible a los expertos humanos, incluso si este reordenamiento no mejora la eficiencia en la solución de problemas.

Los SBCs logran un desempeño mayor si dentro de sus funciones está:

- Poder evitar errores cometidos anteriormente y anticipar problemas que han ocurrido previamente.
- Poder acortar el proceso de solución de problemas, adaptando y almacenando el nuevo caso según sus características, y así utilizarlo para resolver otros problemas similares disminuyendo la cantidad de pasos a realizar y no tener que comenzar este proceso desde sus inicios.
- Ofrecer mejores soluciones, ya que con el paso del tiempo pueden inferirse mejores respuestas que las que hubiera podido dar con menos experiencia. La BCs almacena casos que contemplan tanto fracasos como casos exitosos y así será más completo que uno cuya base sólo contenga situaciones exitosas.

La posibilidad de aprendizaje de los SBCs está en la acumulación y generación automática de nuevos casos el perfeccionamiento y reorganización de la BCs.

1.4.2 Módulos de los SBCs

1. Base de Casos

Un SBCs no requiere un modelo completo del dominio, pero puede producir diseños completos y complejos, aún con una pequeña base de conocimientos (casos). El diseño comienza por casos completos, que incluyen las conexiones o vínculos entre diversas funciones, esto permite evadir el problema de la optimización de múltiples criterios.

Un caso es un conjunto arbitrario de rasgos usado para describir un concepto particular. Los casos se distinguen de otras formas de conocimiento o experiencia, porque ellos son ejemplos específicos de hechos.

Al comenzar con casos completos se reduce la complejidad y se incrementa la eficiencia al solucionar el problema. El uso de casos como fuente de conocimiento facilita el “aprendizaje” mediante el simple almacenamiento de nuevos casos.

El conjunto de problemas resueltos (casos) forman la BCs o memoria permanente del sistema. En la BCs se registran los casos resueltos; esta debe poseer una organización que le permita mostrar cualidades similares a la memoria humana, es decir, debe ser ilimitada, en la medida que la memoria crezca no se puede hacer más lenta y debe permitir buscar directamente los elementos de memoria que sean relevantes a un problema.

El modelo de memoria puede ir desde usar directamente como memoria una base de datos (relacional, jerárquica, plana) hasta tener una organización propia de los elementos de la misma. Está claro que emplear una base de datos como memoria simplifica el trabajo de crear la misma, pero afecta las propiedades deseadas para una memoria.

Para obtener una organización en la base de datos, la solución debe poseer una estrategia de generalización que cree y mantenga una jerarquía, en la cual se organice la información, que sea incremental, por tanto, a medida que se añadan nuevas instancias a la memoria se realicen mejores generalizaciones.

La base de datos también debe ser pragmática en el sentido de que ningún elemento de la jerarquía se elimine por un simple contraejemplo; cuando se encuentren varias instancias que estén en desacuerdo con la generalización, se deben apartar las partes malas y mantener las buenas; para esto se puede mantener, para cada componente de la generalización, un nivel de confianza que indique cuántas veces este fue confirmado y cuántas veces contradicho.

El modelo de la BCs está estrechamente relacionado con el módulo recuperador de casos e influye en la efectividad de éste.

2. Recuperador de casos

La particularidad del RBC implica que la solución de un problema se realice recuperando, desde la memoria, los problemas ya resueltos almacenados en la BCs que tienen relación con éste. En realidad no existe una equivalencia exacta entre dos problemas, por lo que se requiere de un método para comparar el problema con los ya resueltos registrados en la BCs y seleccionar los que posean una mayor similitud con el mismo. Este es el llamado método de acceso, que utiliza las funciones de semejanza.

Es necesario aclarar que cada vez que se emplee el término “recuperación”, se emplea como sinónimo de rescatar, recobrar, obtener de la BCs.

El mecanismo de recuperación en un SBCs asegura que los casos más relevantes sean recuperados para el problema dado, siendo este el resultado más importante del RBC. Los casos relevantes son aquellos que coinciden con la mayoría de los rasgos del problema presente. La aplicación de la

recuperación está estrechamente relacionada con las aplicaciones de indización y organización de la memoria. [13]

Una vez recuperado el conjunto de elementos candidatos desde la BCs, la selección del caso que mayor grado de semejanza tiene con el nuevo problema puede producir dos resultados: existe uno o varios casos que coinciden totalmente con el patrón de búsqueda, o sea, los valores para los rasgos conocidos son iguales en el caso y en el patrón de búsqueda, o todos los elementos del conjunto de candidatos difieren en al menos un rasgo con el patrón de búsqueda. Si dado un problema se alcanza el primer resultado esto significa que se tienen el o los casos sobre los que se debe trabajar. En la segunda, existen dos alternativas a seguir, una es realizar una búsqueda por analogía y la otra determinar un valor numérico que indique el grado de semejanza o diferencia entre el patrón de búsqueda y cada caso candidato.

Si se realiza la búsqueda por un valor numérico que indique el grado de semejanza entonces el acercamiento a la solución puede hacerse mediante un algoritmo de acceso a la BCs utilizando factores de comparación de los rasgos de semejanza y, finalmente, emplear alguna expresión matemática como función de semejanza.

Una de las funciones de semejanza más utilizada es la siguiente:

$$\beta(P, R) = \sum_{i=1}^n W_i * \delta_i(P_i.v, R_i.v)$$

Donde P representa la información del problema y R representa la información del caso de la BCs. La sumatoria recorre los n rasgos que caracterizan el problema e incluye el producto del peso de cada rasgo W_i con el valor de la función de comparación por cada rasgo δ_i .

Debe garantizarse que: $\sum_{i=1}^n W_i = 1$.

Las funciones de comparación por rasgos más comunes son:

$$a) \delta_i(\alpha, \beta) = \begin{cases} 1 \rightarrow \alpha = \beta \\ 0 \rightarrow \alpha \neq \beta \end{cases}$$

$$b) \delta_i(\alpha, \beta) = \begin{cases} 1 \rightarrow |\alpha - \beta| \leq \varepsilon \\ 0 \rightarrow |\alpha - \beta| > \varepsilon \end{cases}$$

Resulta obvio que la selección del conjunto de rasgos es una cuestión central en el RBC pues incide tanto en la definición del modelo para la memoria permanente como en el método de recuperación de casos. El conjunto de rasgos determina qué información será almacenada en memoria para cada uno de sus elementos, la cual debe permitir la posterior recuperación de los casos semejantes dada la descripción de un nuevo problema.

Para controlar la cantidad de casos sobre los que se trabajará luego de la recuperación y como una vía de realizar la selección, la profesora Janet L. Kolodner [14] propuso el siguiente orden de heurísticas de preferencia:

- I. Preferencia dirigida por objetivo, en la cual se prefieren los casos que direccionan el objetivo de razonamiento actual.
- II. Preferencia basada en rasgos distintivos, en la cual se prefieren los casos en los cuales hay una mayor correspondencia con respecto a los rasgos más importantes en una situación particular.
- III. Preferencia basada en especificidad, prefiere los casos más específicos respecto al problema a resolver.
- IV. Preferencia basada en frecuencia, prefiere los casos más frecuentemente recuperados.
- V. Preferencia basada en el tiempo de recuperación, en la cual se prefieren los casos más recientemente recuperados.
- VI. Preferencia basada en la facilidad de adaptación, en la cual se prefieren los casos con rasgos que son más fácilmente adaptados a la nueva situación.

Resulta obvio que la selección del conjunto de rasgos es una cuestión central en el RBC pues incide tanto en la definición del modelo para la memoria permanente como en el método de recuperación de casos. El conjunto de rasgos determina qué información será almacenada en memoria para cada uno de sus elementos, la cual debe permitir la posterior recuperación de los casos semejantes dada la descripción de un nuevo problema.

La comparación de la descripción del problema a resolver y que se utiliza como patrón de búsqueda con cada elemento de la memoria permanente se realiza rasgo a rasgo. Para esto es necesario que una vez que se hayan seleccionado los rasgos, para cada uno se determine su dominio de definición y seguidamente el criterio de comparación de sus valores (cuándo son diferentes y en qué grado).

Potencialmente en el conjunto de rasgos seleccionados podrían estar todas las propiedades que describen los objetos o elementos del dominio de aplicación, esto indica, todos los rasgos que describen los casos resueltos y que serán almacenados como elementos de la memoria permanente. Sin embargo, generalmente esto no ocurre así. El primer criterio de selección es separar los rasgos inútiles.

3. Adaptación

Después de seleccionados y recuperados los casos más parecidos, se procede a elaborar el nuevo, modificándolo de ser necesario. Este nuevo caso obtenido como resultado final será incorporado a la BCs como uno más, lo que permitirá que el sistema se desarrolle “aprendiendo” de sí mismo. En la implementación de los métodos de adaptación se hace imprescindible el conocimiento de los expertos en este dominio.

La adaptación de un caso puede realizarse, ya sea por la transformación de un caso solo para ajustarlo a los requisitos de la nueva situación, o mediante la unión apropiada de partes de varios casos. La adaptación a menudo es muy difícil y puede ser eludida del todo, es por esto que ella ha recibido menos atención que la recuperación. Sin embargo, este es un problema central para el diseño de un SBCs, a pesar que hay sistemas que no tienen adaptador.

La evaluación y modificación de los métodos de adaptación son importantes para construir un sistema eficaz. Si un SBCs puede corregirse a sí mismo constantemente mediante la evaluación, se dice que es adaptativo.

El sistema sólo presentará las experiencias relevantes al usuario y este es quien realiza la adaptación. La solución al nuevo problema es resultante de un proceso de cooperación en el cual la máquina tiene almacenados y recupera los casos y la adaptación la realiza el experto humano.

Ciclo de vida de RBC

En el proceso de dividir el RBC en diferentes subprocessos nos encontramos que el ciclo que lo conforma puede ser dividido en cuatro procesos claramente diferenciados:

- RECUPERAR los casos similares al que se está analizado.
- REUTILIZAR la información y el conocimiento que se tiene de este caso para resolver el problema.
- REVISAR la solución propuesta.
- RETENER las partes de esta experiencia que puedan ser útiles para la solución de futuros problemas.

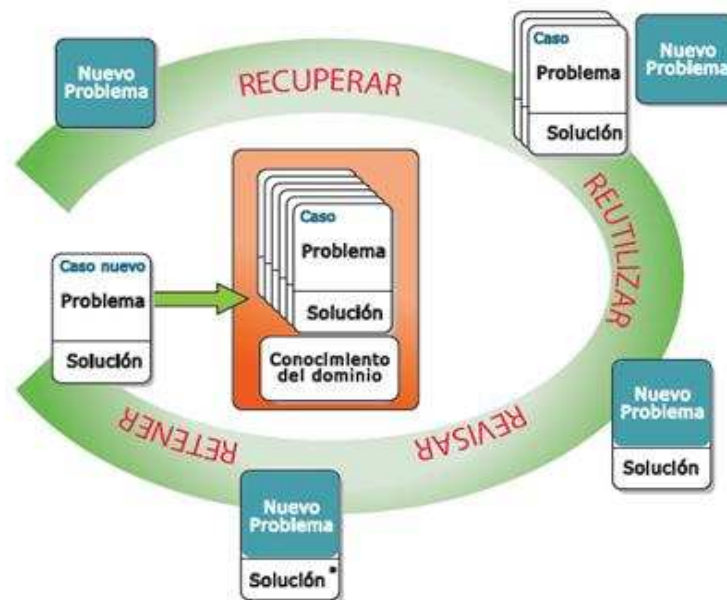


Fig.3 Ciclo del Razonamiento Basado en Casos.

Hacer cualquier inferencia basada en casos incluye los pasos siguientes:

- Presentar como entrada al sistema una descripción del problema a resolver o problema actual (Presentación).
- Encontrar en memoria aquellos casos relevantes que resuelven problemas similares al actual (Recuperación).
- Determinar cuáles partes del viejo caso se deben focalizar para resolver el sub-problema actual.
- Derivar una solución adaptando la solución previa a las restricciones del nuevo problema (Adaptación).
- Chequear la consistencia del valor derivado con la descripción del problema a resolver, y aceptar o rechazar este (Validación).
- Si se considera apropiada la solución validada se añade a la BCs para ser usada en el futuro si se considera conveniente (Actualización).

Finalmente, resolver problemas adaptando soluciones conocidas permite al resolvidor de problemas no tener que considerar muchas restricciones, pues solamente hay que tener en cuenta aquellas que se relacionan con los elementos a adaptar. Algunos sistemas mantienen casos casi usables y casos hipotéticos para mejorar el proceso de adaptación, entre ellos HYPO y XBE. Los casos hipotéticos son incompletos pero introducen nuevas opciones de solución.

1.5 SBC existentes

A nivel internacional:

- **e2glite**: es un software que se distribuye gratuitamente y que permite manejar razonamiento con incertidumbre a través de Factores de confianza (CF). E2gLite es un archivo Java que contiene el motor de inferencias y se carga como un applet mediante etiquetas especiales contenidas en una página Web. De esta forma, cualquier browser que tenga instalado la JVM (Máquina virtual java) puede ser utilizado como interfaz de usuario. La base de conocimientos de ContractLite reside en un archivo que es pasado como parámetro al applet. E2glite posee un lenguaje sencillo pero apropiado para la construcción de sistemas basados en conocimientos sobre la Web.

- **Kayako SupportSuite:** producto insignia de apoyo SupportSuite es robusto y flexible, que permite implementar efectivamente los canales de apoyo, e-mail y administrar la gestión de auto-ayuda de recursos. SupportSuite hace esto mediante la combinación de ticket de apoyo (basado en Web y e-mail), chat y una intuitiva interfaz de cliente.
- **Neocenter:** Base de conocimientos utilizado para gestionar el conocimiento de las plantas telefónicas. Es una aplicación desarrollada en plataforma Web. Presenta un buscador de artículos referentes a este dominio y especializado por categorías. Permite consultar artículos tecnológicos de interés para técnicos y especialistas.
- **Arandasoft:** Una base de conocimientos que proporciona diferentes opciones para consultar el contenidos de interés. Presenta un navegador donde es posible visualizar los últimos artículos relevantes, un buscador que implementa una búsqueda avanzada y un sistema de preguntas y respuesta más frecuentes.

A nivel nacional:

En Cuba se han realizado trabajos científico-investigativos sobre los SBCs en distintas universidades del país: ISPJAE, UCLV, entre otros centros, abordando diversos temas dentro del campo de la Ingeniería Mecánica y la Ingeniería de Software.

A nivel institucional:

En la UCI se trabaja de forma independiente en la creación o modificación de algunas fuentes de software libre de sistemas de ticket, pero éstos no incorporan RBC y no satisfacen las necesidades de la investigación.

Para cumplir con los objetivos específicos de la investigación se determinó que lo adecuado es desarrollar un SBCs que permita gestionar los incidentes tecnológicos en la UCI.

1.6 Tecnología, técnicas y herramientas.

La tecnología, técnicas y herramientas a utilizar para desarrollar la solución informática a la problemática planteada, son las propuestas por la DIN en las líneas bases de la arquitectura para el desarrollo de sistemas informáticos en la UCI. Ellas son:

- Las aplicaciones Web como estándar para la Gestión de Servicios en la UCI.

- El patrón Modelo, Vista, Controlador, la separación de la lógica de negocios, presentación y acceso a datos, permitiendo flexibilidad y facilidad si se tienen que hacer cambios futuros.
- PHP como tecnología aprobada por lo antes expuesto, será estandarizado para los sitios Web y Portales por ser más liviano y fácil de utilizar.
- Codeigniter como framework para la confección del sitio Web donde se hosteará esta aplicación.
- Ajax como tecnología del lado del cliente para acelerar las peticiones al servidor de Base de Datos en las búsquedas.
- PostgreSQL como Gestor de Base de Datos. "
- Visual Paradigm como herramienta CASE para la modelación visual.
- SXP como metodología de desarrollo de software.

1.6.1 Uso de Metodologías Ágiles:

Se decide optar por una metodología ágil para el desarrollo del software debido al poco tiempo de desarrollo y el reducido grupo de trabajo. Los principales objetivos que persigue esta metodología son:

- Valorar al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.
- Desarrollar software que funcione más que conseguir una buena documentación.
- La colaboración con el cliente más que la negociación de un contrato.
- Responder a los cambios más que seguir estrictamente un plan.

Los principales valores que conlleva a utilizar estos métodos se expresan en el "Manifiesto Ágil [15]":

- La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- El software que funciona es la medida principal de progreso.
- La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.

- En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

¿Qué es Scrum?

- Define un marco para la gestión de proyectos.
- Indicada para proyectos con un rápido cambio de requisitos.
- El desarrollo de software se realiza mediante iteraciones, denominadas sprints.
- Reuniones a lo largo proyecto.

¿Qué es XP?

- Centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software.
- Basada en continua realimentación entre el cliente y el equipo de desarrollo.
- Simplicidad en las soluciones implementadas.
- Adecuada para proyectos con requisitos imprecisos y muy cambiantes.

Se explica la dinámica del proyecto a través de historias de usuarios, prototipos de interfaz de usuario y otros modelos auxiliares. Está dividida en cuatro fases, que son precisamente la base de la estructura del nuevo expediente de proyecto, estas son:

- Planificación-Definición.
- Desarrollo.
- Entrega.
- Mantenimiento.

Cada una de estas fases está compuesta por una serie de actividades que son las que generan los artefactos que quedan incluidos en el nuevo expediente de proyecto, estas actividades están recogidas en el guión de la metodología. **Ver Anexo 2.**

Conclusiones

El desarrollo y la utilización de sistemas informáticos para la gestión del conocimiento en la actualidad, y los SBC como herramientas tecnológicas desarrolladas con este fin, son entre otras, temáticas abordadas en este capítulo.

La experiencia de expertos en el dominio de las tecnologías, y la gestión de servicios de tecnologías informáticas según normas y metodologías actuales conllevan al estudio de los SBCs, su incidencia en las TIC y utilización en la gestión de incidentes tecnológicos a partir de características y funcionalidades.

El razonamiento basado en casos como técnica de la IA, en la simulación del razonamiento humano, a partir de experiencias adquiridas por estos y la memoria humana como factor fundamental en este proceso.

A partir de la investigación realizada, incluyendo entrevistas a personalidades expertas en materia de IA, se determinó que el SBCs es el tipo de SBC con el que puede implementarse la primera línea de ayuda o nivel "0" en la gestión de incidencias tecnológicas en la DGT de la UCI, pues los SBCs se basan en la experticia y el conocimiento heurístico de expertos en el dominio.

Capítulo 2

Descripción del Sistema.

Introducción

Se describen los objetivos estratégicos de la DGT, el flujo actual de procesos y el objeto de automatización. Se detallará la fase de planificación y definición del sistema, primera fase definida por la metodología SXP, a partir de los artefactos que se generan en la misma.

2.1 Objetivos estratégicos

La DGT se ha trazado como estrategia para mejorar la calidad y efectividad en los servicios implementar a través del GAT una ayuda en primera línea, para así lograr disminuir los problemas tecnológicos y para esto planea brindar mediante esta primera línea de soporte una ayuda que permita disminuir el impacto de los incidentes en las tecnologías.

2.2 Flujo actual de proceso

Actualmente el proceso de gestión de incidentes tecnológicos se brinda a partir de una segunda línea de ayuda donde el personal técnico atiende cada incidente surgido en su área de atención.

Este proceso se inicia cuando un usuario se comunica con el centro de reportes que implementa el GAT, aquí lo atiende la recepcionista en turno, recogiendo los datos del usuario y del incidente. Otro punto importante es el conocimiento que estas posean, ya que por su reciente ingreso, no son expertas en el dominio de los incidentes tecnológicos. En muchas ocasiones el técnico de guardia atiende estos llamados y para dar una solución primaria implementa por vía telefónica una primera línea de ayuda de forma empírica y esta en muchas ocasiones resulta eficiente, y el usuario queda complacido.

2.3 Objeto de automatización

Para poder implementar la primera línea de ayuda en la gestión de incidentes tecnológicos se determinó realizar una base de conocimientos con razonamiento basado en casos a través de una aplicación Web, con el objetivo de almacenar casos que resuelvan los problemas más frecuentes y que permitan ofrecer una ayuda primaria, donde no sea necesario la presencia de un técnico o experto.

El sistema permitirá mediante una interfaz funcional seleccionar los rasgos más representativos de un nuevo problema, y así permitir realizar una inferencia óptima basada en los rasgos descritos por el usuario. Esta funcionalidad se ve representada en el módulo recuperador de casos. Para la gestión de casos resueltos se implementa un formulario que permita al experto almacenar en una base de datos relacional los rasgos que identifiquen o describan el caso.

2.4 Modelo de Dominio

El modelo historias de usuario del negocio, es un artefacto que se genera del juego de la planificación, luego de estar definida la concepción del sistema se hace mucho más fácil comprender el negocio.

En esta plantilla se definen las características específicas del negocio, así como la forma en que interactúa el sistema con los clientes y viceversa. El modelo de negocio cuando se trabaja con metodologías ágiles, es diferente al ya conocido en el proceso unificado, ya que en este caso se trabaja con historias de usuarios, en vez de con casos de uso. Pero independientemente de los cambios técnicos que puedan existir, el negocio se modela igual en cualquier metodología.

Si el negocio no está bien definido, entonces es generado el llamado modelo de dominio. El análisis orientado a objetos tiene por finalidad estipular una especificación del dominio del problema y los requisitos desde la perspectiva de la clasificación por objetos, además desde el punto de vista de entender los términos empleados en el dominio. Para descomponer el dominio del problema hay que identificar los conceptos (unidades comprensibles), atributos y las asociaciones del dominio que se califican importantes. El resultado puede expresarse en un modelo conceptual.

Conceptos y entidades principales.

Usuario: persona que juega un determinado rol para ejecutar una determinada funcionalidad en una aplicación que pertenece a la solución informática.

Problema: síntoma que representa una interrupción en los servicios.

Solución: solución dada por técnicos y expertos para restablecer el servicio.

Caso: experiencias vividas por técnicos y expertos en la solución de problemas.

Equipo: dispositivo con el cual el usuario se beneficia de los servicios que brinda.

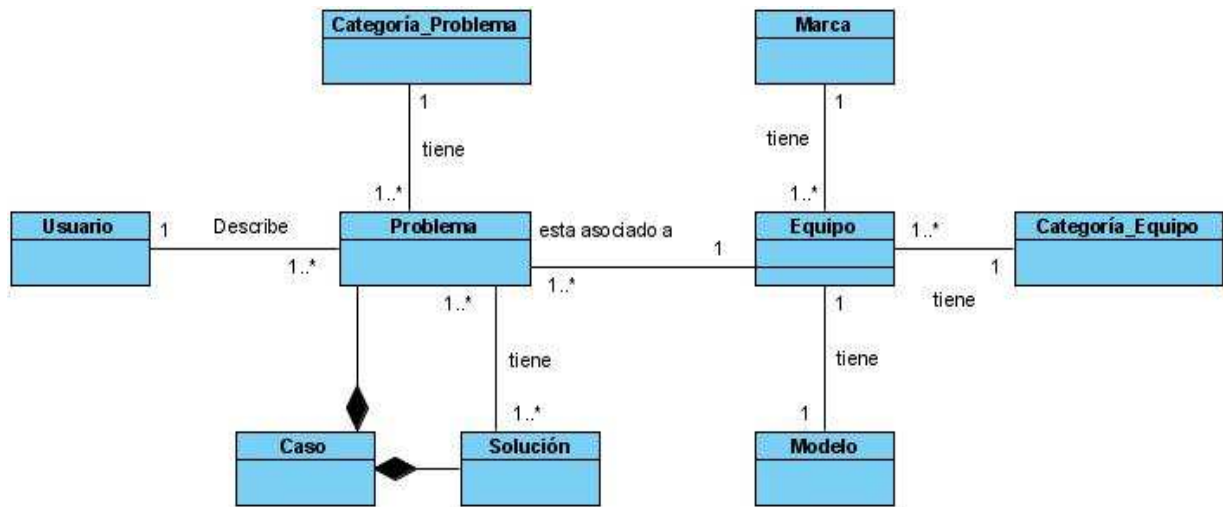


Fig. 4 Modelo de dominio.

2.5 Lista de Reserva del Producto (LRP)

Es una lista priorizada que define el trabajo que se va a realizar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requisitos sobre el producto. Sin embargo, suelen surgir los más importantes que casi siempre son más que suficientes para un Sprint (Iteración).

Esta lista puede crecer y modificarse a medida que se obtiene más conocimiento acerca del producto y del cliente. Con la restricción de que solo puede cambiarse entre Sprint. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto.

2.5.1 Funcionalidades del sistema

Prioridad

- | | | |
|---|--|------|
| 1 | Autenticar usuario. | Alta |
| 2 | Buscar caso dado criterios de búsqueda. | Alta |
| 3 | Mostrar casos dado un problema devueltos por el motor de inferencia. | Alta |
| 4 | Mostrar pasos para la solución. | Alta |
| 5 | Aumentar ranking. | Alta |

6	Adaptar solución a nuevo problema.	Alta
7	Proponer solución de otro caso semejante.	Alta
8	Gestionar datos del caso.	Alta
	8.1. Modificar datos del caso.	
	8.2. Eliminar caso.	
	8.3. Adicionar caso a la base de conocimiento.	
	8.4. Consultar caso.	
9	Generar informe.	Media
10	Gestionar tipo de equipo.	Media
	10.1. Modificar tipo equipo.	
	10.2. Adicionar tipo equipo.	
	10.3. Consultar tipo equipo.	
11	Gestionar marcas.	Media
	11.1. Modificar marcas.	
	11.2. Adicionar marcas.	
	11.3. Consultar marcas.	
12	Gestionar modelo.	Media
	8.1. Modificar modelo.	
	8.2. Adicionar modelo.	

8.3. Consultar modelo.

13 Gestionar síntoma. Media

13.1. Modificar síntoma.

13.2. Adicionar síntoma.

13.3. Consultar síntoma.

14 Gestionar categoría de tipo de equipo. Media

14.1. Modificar categoría de tipo de equipo.

14.2. Adicionar categoría de tipo de equipo.

14.3. Consultar categoría de tipo de equipo.

15 Gestionar categoría síntoma. Media

15.1. Modificar categoría síntoma.

15.2. Adicionar categoría síntoma.

15.3. Consultar categoría síntoma.

16 Gestionar categoría problema. Media

16.1. Modificar categoría problema.

16.2. Adicionar categoría problema.

16.3. Consultar categoría problema.

17 Gestionar pasos.

17.1. Modificar pasos.

17.2. Adicionar pasos.	Alta
17.3. Consultar pasos.	
18 Gestionar usuario.	Alta
18.1. Modificar usuario.	
18.2. Adicionar usuario.	
18.3. Consultar usuario.	
19 Gestionar roll.	Media
19.1. Modificar roll.	
19.2. Consultar roll.	
19.3. Adicionar roll.	
20 Aprobar casos.	Alta

2.5.2 Requisitos no funcionales del sistema

1 Requisitos de Usabilidad.

1.1 El producto puede ser usado por personas con un mínimo de conocimientos sobre los entornos Web y el manejo del ordenador.

2 Requisitos de Seguridad.

2.1 La información que se almacenará en el sistema está protegida de accesos y cambios no autorizados.

2.2 Se mantiene la integridad de todos los datos.

2.3 La información manejada por el sistema esta protegida de acceso no autorizado y divulgación.

3 Requisitos de apariencia o interfaz externa

3.1. Interfaz fácil de manejar por el usuario.

3.2 Se debe evitar recargar las páginas con textos, imágenes o gráficos.

3.3 Cada rol tendrá una interfaz diferente con las acciones que este pueda realizar en el sistema.

4 Requisitos de Rendimiento:

4.1. El tiempo de respuestas a solicitudes en el sistema, debe ser en el menor tiempo posible.

4.2. El producto debe tener gran velocidad de procesamiento ya que el sistema utiliza grandes volúmenes de datos e información.

4.3 El sistema debe estar disponible en todo momento para todos los usuarios

5 Requisitos de Soporte:

El producto requiriere de constantes mantenimientos debido a futuras actualizaciones y cambios, después de instalado.

6 Requisitos de Portabilidad

El sistema es compatible con los sistemas operativos, Linux y Windows, sin

necesidad de cambios significativos.

7 Requisitos de Confiabilidad

En cada mantenimiento se deberán reportar posibles fallos y cómo serán monitoreados en caso de que ocurran.

8 Requisitos de Software

8.1. Se deberá contar con: Navegador de Internet (Mozilla Firefox).

8.2. Para el funcionamiento del sistema en las terminales cliente es necesario el S.O. Windows XP o superior, Linux o Unix.

8.3 Servidor Web Apache 2.0 o superior, y debe estar configurado con la extensión pgsqI incluida.

9 Requisitos de Hardware

Servidor:

9.1 Microprocesador: Intel Pentium III o superior, 800 MHz mínimo. Memoria RAM: 512 MB o superior. Capacidad del disco duro: 80 GB o superior.

PC Cliente:

9.2 Microprocesador: Intel Pentium III o superior, 450 MHz mínimo. Memoria RAM: 128 MB mínimo.

10 Restricciones en el diseño y la implementación.

10.1 Para el análisis y el diseño del sistema debe ser utilizada la metodología

SXP.

10.2 Usando el lenguaje de modelación *UML* y como herramienta para llevarlo a cabo el Visual Paradigm.

10.3 Implementación del sistema en el lenguaje *PHP*.

10.4 *EMS SQL Manager for PostgreSQL*, aplicación cliente utilizada para la gestión de la información en la base de datos.

10.5 *Zend Studio* como herramienta de desarrollo.

2.6 Plan de Releases

Una vez obtenida la concepción del sistema, definidas dichas funcionalidades y agrupadas por historias de usuarios se define el plan de *release* donde se muestran las iteraciones para realizar las entregas intermedias y final de cada funcionalidad del sistema. De acuerdo a la prioridad definida por funcionalidades e interés del cliente se realizó la siguiente planificación:

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total (Semanas)
1	Priorizar la gestión de los nomencladores	10,11,12,13,14,15,16	2
2	Implementar el módulo recuperador de casos. Seguridad mediante autenticación	1,2,3,4,8,17	2
3	Recuperar, adaptar solución.	5,6,18,19,20	1
4	Aprobación por expertos de nuevos casos creados, propuesta de nuevos casos y generar informes	7,9	1

2.7 Historias de usuarios

Las historias de usuarios son la técnica utilizada en XP para especificar los requisitos del software, lo que equivale a los casos de uso en el proceso unificado. Las mismas son escritas por los clientes como las tareas que el sistema debe hacer y su construcción depende principalmente de la habilidad que tenga el cliente para definir las. Son escritas en lenguaje natural, no excediendo su tamaño de unas pocas líneas de texto. Las historias de usuario guían la construcción de las pruebas de aceptación, elemento clave en XP (deben generarse una o más pruebas para verificar que la historia ha sido correctamente implementada) y son utilizadas para estimar tiempos de desarrollo. En este sentido, sólo proveen detalles suficientes para hacer una estimación razonable del tiempo que llevará implementarlas. En el momento de implementar una historia de usuario, se debe detallar a través de la comunicación con el cliente. Estas son la base para las pruebas funcionales.

En esta plantilla los campos de puntos estimados y puntos reales se llenan, luego del desarrollo de la actividad de estimación de esfuerzo, donde se decide que tiempo se le dedicará a cada historia de usuario definida con anterioridad. Las historias de usuario proporcionan ventajas pues:

- Están escritas en lenguaje del cliente, por lo que es muy fácil su comprensión.
- Especifican cada uno de los requisitos del sistema, sin necesidad de documentaciones extensas.
- Reflejan todas las características del sistema.
- Si se definen correctamente, guían el proceso de implementación. **Ver Anexo 3.**

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Autenticar usuario.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Orlando Cruz Rojas- Claudio Puig González	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 2 Semanas
Riesgo en Desarrollo: Medio	Puntos Reales: 2 Semanas
Descripción: El usuario introduce su nombre de usuario y contraseña del dominio para acceder a los módulos de gestión del conocimiento y administración. De acuerdo a los permisos que tenga podrá interactuar con el sistema.	
Observaciones: El sistema tiene los roles administrador, experto y técnico.	
Prototipo de interface:	

BASE DE CONOCIMIENTOS TECNOLÓGICOS



Historia de Usuario

Número: 2	Nombre Historia de Usuario: Buscar caso dado criterios de búsqueda.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Orlando Cruz Rojas- Claudio Puig González	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 2 Semanas
Riesgo en Desarrollo: Medio	Puntos Reales: 2 Semanas
Descripción: El usuario selecciona los rasgos asociados al problema (equipo, marca, modelo, síntoma y categoría del problema), y a partir de esta información se realiza la inferencia.	
Observaciones: A partir de experiencias vividas por los desarrolladores y un estudio realizado se definieron rasgos objetivos (equipo, marca, modelo, síntoma y categoría del problema) a los cuales se les asignó un peso de acuerdo al resultado del estudio realizado.	
Prototipo de interface:	

Recuperador de casos

Datos del equipo

Equipo: Marca: Modelo:

Datos del problema

Sintoma: Categoría:

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Mostrar casos dado un problema devuelto por el motor de inferencia.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Orlando Cruz Rojas- Claudio Puig González	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 2 Semanas
Riesgo en Desarrollo: Medio	Puntos Reales: 2 Semanas
Descripción El sistema mostrará los casos existentes de acuerdo a la categoría del equipo y del síntoma. Además a partir del peso asignado y el valor de similitud devuelto de la comparación de los atributos, se realizará la inferencia mostrando el grado de semejanza entre el nuevo caso y los existentes en la base de casos.	
Observaciones: Para establecer la semejanza entre el problema actual y los ya resueltos, el MI utiliza una fórmula de semejanza en la cual el peso de los rasgos de igual similitud tiene gran importancia.	

Prototipo de interface:

Recuperador de casos

Datos del equipo

Equipo: Marca: Modelo:

Datos del problema

Síntoma: Categoría:


Mostrar registros

Pasos	Equipo	Desc. Problema	Desc. Solución	Ranking	Grado de semejanza
<input type="checkbox"/>	PC	NO ENCIENDE.	COMPROBAR ALIMENTACIÓN	1	0.9
<input type="checkbox"/>	PC	Fuera del Dominio	AGREGAR AL DOMINIO	4	0.5
<input type="checkbox"/>	PC	No tiene OFFICE instalado.	INSTALAR PAQUETE DE OFFICE.	1	0.5

Mostrando desde 1 hasta 3 de 3 registros

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: Historia 3.
Nombre Tarea: Implementar el Motor de Inferencia.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 2 Semanas
Fecha Inicio: 11/03/09	Fecha Fin: 26/03/09
Programador Responsable: Orlando Cruz-Claudio Puig	
<p>Descripción: Realizar un estudio detallado sobre los rasgos objetivos a la hora encuestar al usuario cuando este reporta una incidencia, para así poderles asignar a estos rasgos el peso correcto para que la fórmula de semejanza devuelva los casos que realmente son semejantes.</p>	

Historia de Usuario																									
Número: 4	Nombre Historia de Usuario: Mostrar pasos para la solución.																								
Modificación de Historia de Usuario Número: ninguna																									
Usuario: Orlando Cruz Rojas- Claudio Puig González	Iteración Asignada: 2																								
Prioridad en Negocio: Alta	Puntos Estimados: 2 Semana																								
Riesgo en Desarrollo: Medio	Puntos Reales: 2 Semana																								
Descripción: El sistema muestra los pasos definidos para dar solución del problema según el caso seleccionado.																									
Observaciones: Los pasos se mostrarán en el mismo orden que se introdujeron por el experto.																									
Prototipo de interface:																									
<p style="text-align: center;">Recuperador de casos</p> <p style="text-align: center;">Datos del equipo</p> <p style="text-align: center;">Lea los pasos uno a continuación del otro sin saltar el orden</p> <table border="1"> <thead> <tr> <th>Identificador</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Ir a Mi PC clic derecho Propiedades/Nombre de Equipo</td> </tr> <tr> <td>2</td> <td>Verificar que el nombre se corresponda con el local donde está ubicada la PC (ej. apto13201).</td> </tr> <tr> <td>3</td> <td>Si no corresponde clic Cambiar... y asignar nombre al equipo.</td> </tr> <tr> <td>4</td> <td>Windows pedirá reiniciar.</td> </tr> <tr> <td>5</td> <td>Llamar al nodo central (8002</td> </tr> <tr> <td>6</td> <td>8003) y pedir que saquen el nombre que asignamos del dominio.</td> </tr> <tr> <td>7</td> <td>Nuevamente Mi PC clic derecho Propiedades/Nombre de Equipo/ Cambiar...</td> </tr> <tr> <td>8</td> <td>Seleccionar Dominio y poner uci.cu</td> </tr> <tr> <td>9</td> <td>clic Aceptar.</td> </tr> <tr> <td>10</td> <td>Windows pedirá su usuario-contraseña del dominio.</td> </tr> <tr> <td>11</td> <td>Clic Aceptar y reiniciar la PC.</td> </tr> </tbody> </table> <p style="text-align: center;">¿Resolvio el problema?</p> <p style="text-align: center;"> <input type="button" value="Si"/> <input type="button" value="No"/> </p>		Identificador	Descripción	1	Ir a Mi PC clic derecho Propiedades/Nombre de Equipo	2	Verificar que el nombre se corresponda con el local donde está ubicada la PC (ej. apto13201).	3	Si no corresponde clic Cambiar... y asignar nombre al equipo.	4	Windows pedirá reiniciar.	5	Llamar al nodo central (8002	6	8003) y pedir que saquen el nombre que asignamos del dominio.	7	Nuevamente Mi PC clic derecho Propiedades/Nombre de Equipo/ Cambiar...	8	Seleccionar Dominio y poner uci.cu	9	clic Aceptar.	10	Windows pedirá su usuario-contraseña del dominio.	11	Clic Aceptar y reiniciar la PC.
Identificador	Descripción																								
1	Ir a Mi PC clic derecho Propiedades/Nombre de Equipo																								
2	Verificar que el nombre se corresponda con el local donde está ubicada la PC (ej. apto13201).																								
3	Si no corresponde clic Cambiar... y asignar nombre al equipo.																								
4	Windows pedirá reiniciar.																								
5	Llamar al nodo central (8002																								
6	8003) y pedir que saquen el nombre que asignamos del dominio.																								
7	Nuevamente Mi PC clic derecho Propiedades/Nombre de Equipo/ Cambiar...																								
8	Seleccionar Dominio y poner uci.cu																								
9	clic Aceptar.																								
10	Windows pedirá su usuario-contraseña del dominio.																								
11	Clic Aceptar y reiniciar la PC.																								

Historia de Usuario	
Número: 5	Nombre Historia de Usuario: Aumentar ranking.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Orlando Cruz Rojas- Claudio Puig González	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1 Semana
Riesgo en Desarrollo: Medio	Puntos Reales: 1 Semana
<p>Descripción: Se realiza una encuesta al usuario si resolvió o no el problema. En caso de ser afirmativa la respuesta y el problema es igual al descrito por el usuario este aumenta el valor en una unidad del atributo ranking de la solución propuesta.</p>	
<p>Observaciones: Este atributo en la solución se irá incrementando por ser una solución que ha dado mayor aporte para resolver problemas y a su vez cuando se realice la inferencia, además del grado de semejanza el valor de este atributo influirá para que aparezca entre los primeros casos.</p>	
<p>Prototipo de interface:</p> 	

Historia de Usuario																															
Número: 6	Nombre Historia de Usuario: Adaptar solución a nuevo problema.																														
Modificación de Historia de Usuario Número: ninguna																															
Usuario: Orlando Cruz Rojas- Claudio Puig González	Iteración Asignada: 3																														
Prioridad en Negocio: Alta	Puntos Estimados: 1 Semana																														
Riesgo en Desarrollo: Medio	Puntos Reales: 1 Semana																														
<p>Descripción: El problema está compuesto por el equipo y el síntoma. El usuario describe un problema a partir de los rasgos seleccionados. Al seleccionar un caso para ver los pasos, si resuelve el problema se reorganiza la base de casos creando un nuevo problema, asignándole la solución propuesta.</p>																															
<p>Observaciones: Un problema puede tener varias soluciones y viceversa.</p>																															
<p>Prototipo de interface:</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p style="text-align: center;">Recuperador de casos</p> <p style="text-align: center;">Datos del equipo</p> <p>Equipo: <input type="text" value="MOUSE"/> Marca: <input type="text" value="Seleccione"/> Modelo: <input type="text" value="Seleccione"/></p> <p style="text-align: center;">Datos del problema</p> <p>Síntoma: <input type="text" value="PC no lo reconoce"/> Categoría: <input type="text" value="SOFTWARE"/></p> <p style="text-align: center;"><input type="button" value="Recuperar"/></p> <p>Mostrar <input type="text" value="10"/> registros Buscar: <input type="text"/></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Pasos</th> <th>Equipo</th> <th>Desc. Problema</th> <th>Desc. Solución</th> <th>Ranking</th> <th>Grado de semejanza</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>MOUSE</td> <td>Fuera del Dominio</td> <td>AGREGAR AL DOMINIO</td> <td>5</td> <td>0.4</td> </tr> <tr> <td><input type="checkbox"/></td> <td>MOUSE</td> <td>Fuera del Dominio</td> <td>COMPROBAR ALIMENTACIÓN</td> <td>3</td> <td>0.4</td> </tr> <tr> <td><input type="checkbox"/></td> <td>MOUSE</td> <td>Sin conexión de red.</td> <td>CAMBIAR NÚMERO IP.</td> <td>2</td> <td>0.3</td> </tr> <tr> <td><input type="checkbox"/></td> <td>MOUSE</td> <td>PC no lo reconoce</td> <td>COMPROBAR ALIMENTACIÓN</td> <td>3</td> <td>0.8</td> </tr> </tbody> </table> <p>Mostrando desde 1 hasta 4 de 4 registros ◀ ▶</p> </div>		Pasos	Equipo	Desc. Problema	Desc. Solución	Ranking	Grado de semejanza	<input type="checkbox"/>	MOUSE	Fuera del Dominio	AGREGAR AL DOMINIO	5	0.4	<input type="checkbox"/>	MOUSE	Fuera del Dominio	COMPROBAR ALIMENTACIÓN	3	0.4	<input type="checkbox"/>	MOUSE	Sin conexión de red.	CAMBIAR NÚMERO IP.	2	0.3	<input type="checkbox"/>	MOUSE	PC no lo reconoce	COMPROBAR ALIMENTACIÓN	3	0.8
Pasos	Equipo	Desc. Problema	Desc. Solución	Ranking	Grado de semejanza																										
<input type="checkbox"/>	MOUSE	Fuera del Dominio	AGREGAR AL DOMINIO	5	0.4																										
<input type="checkbox"/>	MOUSE	Fuera del Dominio	COMPROBAR ALIMENTACIÓN	3	0.4																										
<input type="checkbox"/>	MOUSE	Sin conexión de red.	CAMBIAR NÚMERO IP.	2	0.3																										
<input type="checkbox"/>	MOUSE	PC no lo reconoce	COMPROBAR ALIMENTACIÓN	3	0.8																										

Historia de Usuario

Número: 7

Nombre Historia de Usuario: Proponer solución de otro caso semejante.

Modificación de Historia de Usuario Número: ninguna

Usuario: Orlando Cruz Rojas- Claudio Puig González

Iteración Asignada: 4

Prioridad en Negocio: Alta

Puntos Estimados: 1 Semana

Riesgo en Desarrollo: Medio

Puntos Reales: 1 Semana

Descripción: Los casos devueltos por la inferencia se listarán con un *checkbox* al lado de cada uno. Al hacer clic en el *checkbox* se mostrarán las soluciones, correspondientes al caso seleccionado, en una nueva ventana. Si el caso seleccionado no soluciona el problema, el sistema le propone seguir consultando la solución de otros casos.

Observaciones: Un problema puede tener asociada muchas soluciones y esta a su vez estará dividida en pasos a seguir para lograr la solución.

Prototipo de interface:

Datos del problema

Sintoma: Categoría:

Mostrar registros

Mensaje
Usted puede seguir consultando la solución de otros casos

Pasos	Equipo	Desc. Problema	Desc. Solución	Ranking	Grado de semejanza
<input type="checkbox"/>	IMPRESORA	Fuera del Dominio	AGREGAR AL DOMINIO	5	0.5
<input type="checkbox"/>	MOUSE	Fuera del Dominio	AGREGAR AL DOMINIO	5	0.8
<input type="checkbox"/>	MOUSE	Sin conexión de red.	CAMBIAR NÚMERO IP.	2	0.3
<input type="checkbox"/>	MOUSE	Fuera del Dominio	COMPROBAR ALIMENTACIÓN	2	0.8

Mostrando desde 1 hasta 4 de 4 registros

Historia de Usuario

Número: 8

Nombre Historia de Usuario: Gestionar datos del caso.

Modificación de Historia de Usuario Número: ninguna

Usuario: Orlando Cruz Rojas- Claudio Puig González

Iteración Asignada: 2

Prioridad en Negocio: Alta

Puntos Estimados: 2 Semanas

Riesgo en Desarrollo: Medio

Puntos Reales: 2 Semanas

Descripción: Permite a técnicos y expertos crear casos con los respectivos paso de la solución. Se podrán modificar y eliminar los casos en dependencia del creador, mientras estos no hayan sido aprobados. Solo los expertos tendrán permisos de aprobar casos.

Observaciones: Estos casos son experiencias que el experto a tenido y lo sube a la base casos para hacer explícita esa experiencia.

Prototipo de interface:

DATOS DEL CASO

Mo

Equipo o Periferico

Descripción: PC

Marca: GENERICA

Modelo: GENERICO

Pertenece a: COMPUTO

Gestionar

...

...

...

Síntoma

Descripción: NO ENCIENDE.

Categoría: HARDWARE

Solución

Descripción: comprobar la alimentación

Encienda

Pasos:

- Encienda la PC y compruebe el mouse en Windows.
- Encienda el monitor y la PC.
- Encienda la PC.
- Cuando encienda la PC en la primera pantalla presione la tecla que

Comprueba que la PC este correctame

>>

<<

Guardar

Cancelar

Mo

[Agregar caso](#)

ACTUALIZAR CASO

Equipo: Seleccione ▾
Marca: Seleccione ▾
Modelo: Seleccione ▾
Síntoma: no responden las teclas
Categoría: HARDWARE ▾
Solución: CAMBIO DE TECLADO

Mostrar 10 ▾ registros

Pasos

Editar	Descripción
<input type="checkbox"/>	llevar al taller
<input type="checkbox"/>	cambiar en el local de tránsito

Mostrando desde 1 hasta 2 de 2 registros

Historia de Usuario

Número: 9 **Nombre Historia de Usuario:** Generar informe.

Modificación de Historia de Usuario Número: ninguna

Usuario: Orlando Cruz Rojas- Claudio Puig González **Iteración Asignada:** 4

Prioridad en Negocio: Media **Puntos Estimados:** 1 Semana

Riesgo en Desarrollo: Medio **Puntos Reales:** 1 Semana

Descripción: Se emitirán reportes necesarios para llevar estadísticas acerca de los casos más consultados, los de mayor ranking, entre otros.

Observaciones: Permitirá conocer los mayores problemas existentes, el porcentaje de solución, así como poder implementar la primera línea de ayuda.

Prototipo de interface:

	Informes	Ver
	Informe total de casos	
Total		21
Creados		3
Aprobados		18
	Cerrar	

Historia de Usuario	
Número: 10	Nombre Historia de Usuario: Gestionar tipo de equipo.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Orlando Cruz Rojas- Claudio Puig González	Iteración Asignada: 1
Prioridad en Negocio: Media	Puntos Estimados: 2 Semanas
Riesgo en Desarrollo: Medio	Puntos Reales: 2 Semana
Descripción: Permite al experto insertar, editar y consultar tipos de equipos y las categorías que estos estén asociados.	
Observaciones:	
Prototipo de interface:	

GESTIONAR TIPO EQUIPO

Equipo:

Categoría:

Mostrar registros

Editar <input type="button" value="▲"/>	Descripción <input type="button" value="▼"/>
<input type="checkbox"/>	PC
<input type="checkbox"/>	IMPRESORA
<input type="checkbox"/>	MOUSE
<input type="checkbox"/>	TECLADO
<input type="checkbox"/>	MONITOR

Mostrando desde 1 hasta 5 de 5 registros

GESTIONAR CATEGORÍA

Categoría:

Mostrar registros

Editar <input type="button" value="▲"/>	Descripción <input type="button" value="▼"/>
<input type="checkbox"/>	COMPUTO
<input type="checkbox"/>	PERIFERICO

Mostrando desde 1 hasta 2 de 2 registros

Historia de Usuario	
Número: 11	Nombre Historia de Usuario: Gestionar marca.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Orlando Cruz Rojas- Claudio Puig González	Iteración Asignada: 1
Prioridad en Negocio: Media	Puntos Estimados: 2 Semanas
Riesgo en Desarrollo: Medio	Puntos Reales: 2 Semanas

Descripción: Permite al experto insertar, editar y consultar marcas de equipos.

Observaciones:

Prototipo de interface:

GESTIONAR MARCA

Marca:

Mostrar registros

Editar	Descripción
<input type="checkbox"/>	GENERICA
<input type="checkbox"/>	ASUS

Mostrando desde 1 hasta 2 de 2 registros

Historia de Usuario

Número: 12 **Nombre Historia de Usuario:** Gestionar modelo.

Modificación de Historia de Usuario Número: ninguna

Usuario: Orlando Cruz Rojas- Claudio Puig González **Iteración Asignada:** 1

Prioridad en Negocio: Media **Puntos Estimados:** 2 Semanas

Riesgo en Desarrollo: Medio **Puntos Reales:** 2 Semanas

Descripción: Permite al experto insertar, editar y consultar modelos de equipos.

Observaciones:

Prototipo de interface:

GESTIONAR MODELO

Modelo:

Mostrar registros

Editar	Descripción
<input type="checkbox"/>	GENERICO

Mostrando desde 1 hasta 1 de 1 registros

Historia de Usuario

Número: 13

Nombre Historia de Usuario: Gestionar síntoma.

Modificación de Historia de Usuario Número: ninguna

Usuario: Orlando Cruz Rojas- Claudio Puig González

Iteración Asignada: 1

Prioridad en Negocio: Media

Puntos Estimados: 2 Semanas

Riesgo en Desarrollo: Medio

Puntos Reales: 2 Semanas

Descripción: Permite al experto insertar, editar y consultar síntomas y las categorías asociadas a síntomas.

Observaciones:

Prototipo de interface:

GESTIONAR SÍNTOMA

Síntoma:

Categoría:

Mostrar registros

Buscar:

Editar	Descripción	Categoría
<input type="checkbox"/>	IP contaminado.	Generica
<input type="checkbox"/>	Sin conexión de red.	Generica
<input type="checkbox"/>	NO SE VE	Generica
<input type="checkbox"/>	Fuera del Dominio	Generica
<input type="checkbox"/>	PC no lo reconoce.	Generica
<input type="checkbox"/>	PC no lo reconoce	Generica
<input type="checkbox"/>	NO ENCIENDE.	Generica
<input type="checkbox"/>	No enciende	Generica
<input type="checkbox"/>	Sin Señal	Generica
<input type="checkbox"/>	No tiene OFFICE instalado.	Generica

Mostrando desde 1 hasta 10 de 12 registros

Pasos	Equipo	Sintoma	Solución	Aprobar	Editar	Eliminar	Estado
<input type="checkbox"/>	PC	IP contaminado.	ELIMINAR SPYWARE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	✓ Aprobado
<input type="checkbox"/>	PC	Sin conexión de red.	INSTALAR CONTROLADOR DE RED.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Creado
<input type="checkbox"/>	PC	Sin conexión de red.	ACTIVAR CONEXIÓN DE RED	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Creado

GESTIONAR CATEGORIA SÍNTOMA

Categoría:

Mostrar registros

Editar	Descripción
<input type="checkbox"/>	Generica

Mostrando desde 1 hasta 1 de 1 registros

Historia de Usuario	
Número: 14	Nombre Historia de Usuario: Gestionar categoría problema.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Orlando Cruz Rojas- Claudio Puig González	Iteración Asignada: 1
Prioridad en Negocio: Media	Puntos Estimados: 2 Semanas
Riesgo en Desarrollo: Medio	Puntos Reales: 2 Semanas
Descripción: Permite al experto insertar, editar y consultar categorías asociadas a problema.	
Observaciones:	
Prototipo de interface:	



Historia de Usuario	
Número: 15	Nombre Historia de Usuario: Gestionar pasos.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Orlando Cruz Rojas- Claudio Puig González	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 2 Semanas
Riesgo en Desarrollo: Medio	Puntos Reales: 2 Semanas
Descripción: Permite al experto insertar, modificar y consultar pasos.	
Observaciones:	
Prototipo de interface:	

DATOS DEL CASO

Equipo o Periferico	Gestionar
Descripción: PC	<input style="width: 20px; height: 15px;" type="button" value="..."/>
Marca: GENERICA	<input style="width: 20px; height: 15px;" type="button" value="..."/>
Modelo: GENERICO	<input style="width: 20px; height: 15px;" type="button" value="..."/>
Pertenece a: COMPUTO	

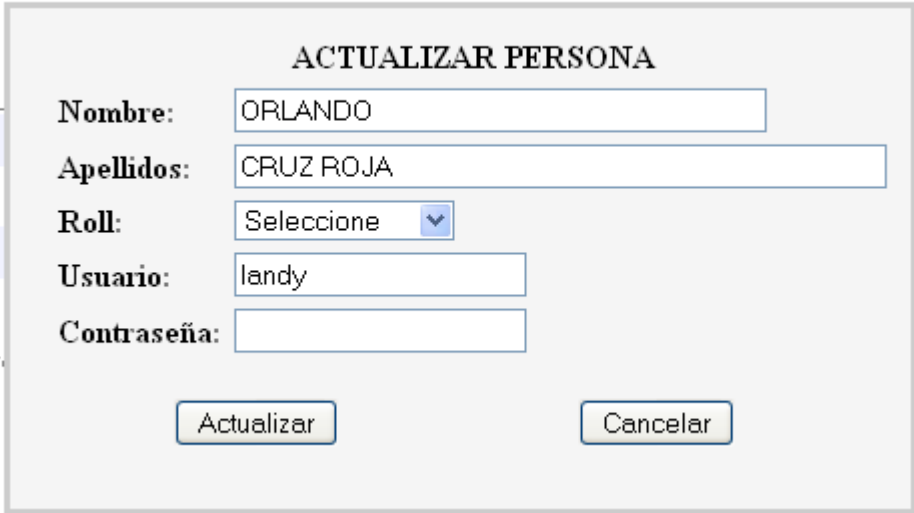
Sintoma	
Descripción: Sin conexión de red.	<input style="width: 20px; height: 15px;" type="button" value="..."/>
Categoría: SOFTWARE	<input style="width: 20px; height: 15px;" type="button" value="..."/>

Solución	
Descripción: activar conexion de red	
Compruebe	<input style="width: 20px; height: 15px;" type="button" value=">>"/>
Pasos:	<input style="width: 20px; height: 15px;" type="button" value="<<"/>
<ul style="list-style-type: none"> Encienda la PC y compruebe el mouse en Windows. Encienda la PC y compruebe el teclado. Compruebe en la parte trasera de la PC si tiene un botón de encen Compruebe el monitor este correctamente conectado a la corriente Compruebe que la impresora este correctamente conectada a la cc 	<div style="border: 1px solid gray; padding: 2px; width: fit-content;"> Ir a Mi PC clic derecho Propiedades/No Verificar que el nombre se correspond </div>
<input type="button" value="Guardar"/>	<input type="button" value="Cancelar"/>

[Agregar caso](#)

Actualizar Paso

Descripción:

Historia de Usuario	
Número: 16	Nombre Historia de Usuario: Gestionar usuario.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Orlando Cruz Rojas- Claudio Puig González	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1 Semana
Riesgo en Desarrollo: Medio	Puntos Reales: 1 Semana
Descripción: Permite al administrador agregar usuarios y asignar los roles de experto o técnico.	
Observaciones:	
Prototipo de interface:  <p style="text-align: center;">GESTIONAR PERSONA</p> <p style="text-align: center;">ACTUALIZAR PERSONA</p> <p>Nombre: <input type="text" value="ORLANDO"/></p> <p>Apellidos: <input type="text" value="CRUZ ROJA"/></p> <p>Roll: <input type="text" value="Seleccione"/></p> <p>Usuario: <input type="text" value="landy"/></p> <p>Contraseña: <input type="text"/></p> <p style="text-align: center;"> <input type="button" value="Actualizar"/> <input type="button" value="Cancelar"/> </p>	

GESTIONAR PERSONA

»S

Nombre

na

ACELYS

LANDO

CLAUDIO

sta 4 de 4

GESTIONAR USUARIO

Nombre:

Apellidos:

Usuario:

Roll:

Password:

Historia de Usuario	
Número: 18	Nombre Historia de Usuario: Aprobar casos.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Orlando Cruz Rojas- Claudio Puig González	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1 Semana
Riesgo en Desarrollo: Medio	Puntos Reales: 1 Semana
Descripción: El experto aprobará los casos que estén correctamente descritos.	
Observaciones:	
Prototipo de interface:	

[Agregar caso](#)

APROBAR CASO

Equipo:
Marca:
Modelo:
Sintoma: (*)
Categoría:
Solución: (*)
Mostrar registros

	Pasos	
Editar		Descripción
<input type="checkbox"/>		llevar al taller
<input type="checkbox"/>		cambiar en el local de tránsito

Mostrando desde 1 hasta 2 de 2 registros ◀ ▶

2.8 Descripción de la Arquitectura

2.8.1 Modelo Vista Controlador

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos:

- Modelo: representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.
- Vista: presenta el modelo en un formato adecuado para interactuar con el usuario.
- Controlador: responde a eventos o acciones del usuario e invoca cambios en el modelo y la vista.

El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la lógica de negocio. En la implementación de la aplicación se hará uso de este patrón arquitectónico.

CodeIgniter está basado en el patrón MVC. En la práctica, permite que sus páginas web contengan mínima codificación ya que la presentación es separada del código PHP. CodeIgniter también le permite incorporar sus códigos existentes, o incluso desarrollar librerías de núcleo para el sistema, habilitándolo a trabajar en una forma que hace que tenga más sentido.

2.8.2 Diseño con metáforas

La plantilla del modelo de diseño, es el artefacto que se genera del diseño con las metáforas, donde se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. En SXP no se enfatiza la definición temprana de una arquitectura estable para el sistema, dicha arquitectura se asume de forma evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora.

El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema, define que la práctica de la metáfora consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema.

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, etc.

Teniendo en cuenta las características anteriores, se define en esta plantilla, un esbozo inicial del diseño del sistema, sin entrar en especificaciones, ni detalles, solo lo que el diseñador necesita para hacer un primer entregable del sistema. Esta plantilla proporciona ventajas, tales como:

- Permite confeccionar un diseño inicial y sencillo del sistema.
- Es la base para la definición de una futura arquitectura.

El sistema está descrito a través del siguiente diagrama de componentes, donde se agrupan los elementos físicos del sistema en paquetes, evidenciándose el patrón de arquitectura MVC.

Las vistas incluyen al subsistema jquery como *framework* de la capa de presentación. Dentro del paquete *controllers* están implementadas todas las funciones que debe realizar el sistema, y este al igual que el paquete *models* interactúan con el *framework* Codeigniter. El componente **usuario_gat.php** interactúa con la librería **uci_ldap.php** que se adiciona como una clase externa de los componentes del *framework*. Dentro del paquete *models* están los componentes que gestionan el acceso a datos a través de *postgre sql*.

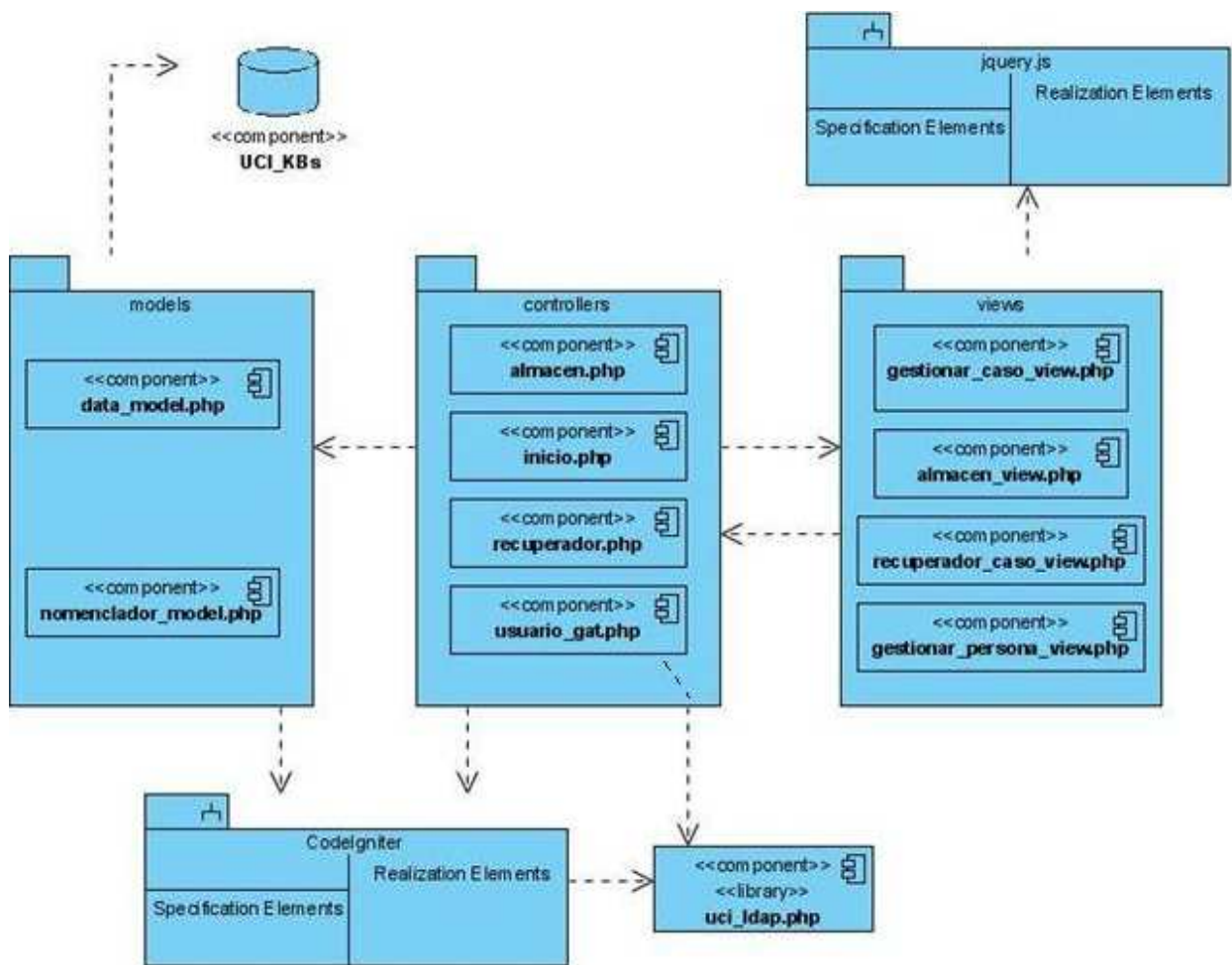


Fig.5 Diagrama de componentes.

2.9 Modelo de datos

Para la persistencia de la información necesaria, y para el correcto funcionamiento del sistema se identificaron 17 tablas.

Los casos están compuestos por soluciones y problemas, estos últimos estarán formados por el equipo y su configuración, así como el síntoma que éstos presenten, agregando una descripción al síntoma junto a un estado asociado. Vale destacar la indexación de los principales rasgos o atributos que permiten optimizar las búsquedas.

Los problemas serán organizados por categorías, para facilitar tanto el almacenamiento como la selección que haga el motor de inferencia. Se podrán agregar categorías en tanto surjan nuevos problemas y éstos necesiten reorganizarse.

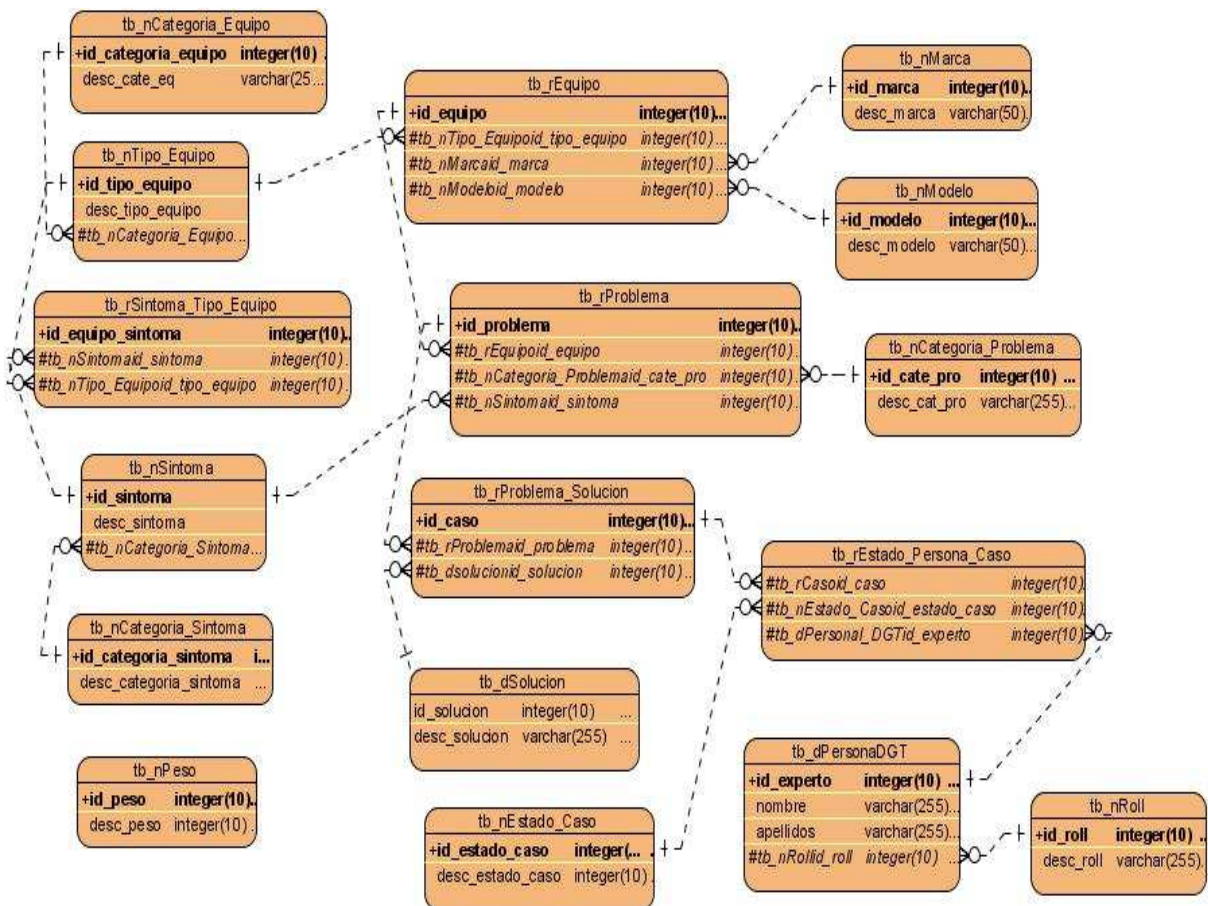


Fig.6 Modelo de datos.

A continuación se explican las principales tablas:

tb_rEquipo: entidad compuesta por la relación de los identificadores de tipo de equipo, marca y modelo.

tb_nTipo_Equipo: entidad donde se almacenarán los tipos de equipos genéricos y la categoría a la que estén asociados.

tb_nSíntoma: entidad donde se almacenarán los síntomas a partir de una descripción y a la categoría asociado.

tb_rProblema: entidad compuesta por la relación de los identificadores del síntoma y el equipo, asociados a categoría.

tb_dSolucion: entidad donde se almacenarán las descripciones de soluciones.

tb_nPaso: entidad donde se almacenarán todos los pasos, descritos y por tuplas.

tb_rSolución_Pasos: entidad donde se almacenarán el identificador de la solución asociado a cada paso para esa solución.

tb_rProblema_Solucion: entidad que almacenará los casos a partir del identificador del problema y la solución correspondiente.

tb_dPersonaDGT: entidad que almacenará los datos de los técnicos y especialistas de la DGT, cada uno de ellos asociados a un roll.

tb_nRoll: entidad nomencladora que almacenará los distintos roles que estarán relacionados con tb_dPersonaDGT.

tb_rEstado_Persona_Caso: entidad donde quedarán registrados todas las incidencias acerca de la creación y aprobación de los casos asociados a creador y quien lo apruebe.

tb_nPeso: entidad donde se almacenarán los atributos y el peso asociado.

Conclusiones

A partir de la concepción inicial del sistema y todas las funcionalidades identificadas por el usuario, se define la LRP, así como la prioridad de cada una de estas, las cuales quedan recogidas en HU, donde se describen los requisitos del sistema.

Se realizaron algunas tareas de ingeniería para lograr una mejor aplicación del motor de inferencia, donde se identificaron los rasgos objetivos y la función de semejanza a implementar.

La planificación de iteraciones se realizó teniendo en cuenta la prioridad para el negocio de cada historia de usuario. Se realizó el diseño de la base de datos del sistema, así como el diagrama de componentes el cual permitió dar una mejor visión del proyecto.

A partir de los objetivos estratégicos definidos en la DGT, el flujo actual de procesos y el objeto de automatización se obtuvo un prototipo no funcional de la solución propuesta.

Capítulo 3

Desarrollo del Sistema.

Introducción

En este capítulo se modelan los artefactos que se obtienen como resultado de la implementación del sistema. Se incluye el código fuente de las principales funcionalidades que realiza el sistema y se brinda una descripción del mismo. Se exponen los casos de pruebas o test de aceptación que se le realizaron al sistema. En este capítulo además de las pruebas se dan a conocer los resultados obtenidos hasta el momento.

3.1 Plantilla código fuente.

En el siguiente código se implementa la fórmula de semejanza, a través de la cual se realiza la inferencia. A partir de los rasgos seleccionados por el usuario para describir el problema (equipo, síntoma, marca, modelo y categoría), se realiza una búsqueda en la BCs de los casos semejantes que estén relacionados con la categoría a la que pertenezca el equipo, así como la marca y modelo; también se tiene en cuenta la categoría definida por el experto del síntoma asociado y la categoría definida por el usuario del problema (hardware o software).

Mediante un ciclo se entra a verificar cada *tupla* o registro devueltos por la función que ejecuta la sentencia SQL. Una vez posicionado en el registro actual se verifica atributo por atributo del caso nuevo con los registros devueltos. A través de otro ciclo *foreach* se recorre cada atributo y se consulta la *tb_nPeso*, donde se almacena los rasgos objetivos con el peso correspondiente. Se comparan los atributos y en caso de que sean iguales se define la variable peso para ese atributo y se multiplica por 1 que es el valor devuelto de la función de comparación entre atributos ($\delta_i(\alpha, \beta) = \begin{cases} 1 \rightarrow \alpha = \beta \\ 0 \rightarrow \alpha \neq \beta \end{cases}$), de lo contrario no se tiene en cuenta el valor devuelto para realizar la inferencia ya que se multiplicaría por '0'.

Una vez recorrido todos los atributos se realiza la suma de las variables de peso y se obtiene la semejanza ($\beta(P,R) = \sum_{i=1}^n W_i * \delta_i(P_i.v, R_i.v)$), de ese caso, tupla o registro.

También se definen variables de estado para agrupar u organizar los resultados obtenidos, en semejanza alta (≥ 0.7), media (≥ 0.5 y < 0.7) y baja (≥ 0.3 y < 0.5), y en ese orden serán mostrados.

En la tabla tb_nPeso se asignaron los siguientes pesos por atributos:

Síntoma (0.4)

Equipo (0.3)

Marca (0.1)

Modelo (0.1)

Categoría (0.1)

```
function buscar_Semejanza()
{
    if ($this->input->post('ajax') == 'true')
    {
        //son del nuevo problema
        $id_tipo_equipoP = $this->input->post('id_tipo_equipo');//id tipo
equipo

        $id_marcaP = $this->input->post('id_marca');// id marca

        $id_modeloP = $this->input->post('id_modelo');// id modelo

        $result_categoria_equipo =
data_model::id_Categoria_Equipo_by_id_Tipo_equipo($id_tipo_equipoP);

        $id_categoria_equipoP = $result_categoria_equipo-
>id_categoria_equipo;

        $desc_sintoma=$this->input->post('desc_sintoma');//desc sintoma

        $result_sintoma =
data_model::id_Sintoma_dado_Desc_Sintoma($desc_sintoma);//id sintoma

        $ouput = '<table class="display" id="example" width="100%"
border="0" cellspacing="2" cellpadding="2" border="1">
```

```
<th align="center"><strong>Pasos</strong></th>
<th align="center"><strong>Equipo</strong></th>
<th align="center"><strong>Desc. Problema</strong></th>
<th align="center"><strong>Desc. Soluci&oacute;n</strong></th>
<th align="center"><strong>Ranking</strong></th>
<th align="center"><strong>Grado de
semejanza</strong></th></thead><tbody>' ;

    if ($result_sintoma)
    {
        $sid_sintomaP = $result_sintoma->id_sintoma;

        $result_categoria_sintoma =
data_model::id_Categoria_Sintoma_dado_Id_Sintoma($sid_sintomaP);

        $sid_categoria_sintomaP = $result_categoria_sintoma-
>id_categoria_sintoma;

        $sid_categoria_problemaP= $this->input-
>post('id_categoria');// id categ de problema

        $data=data_model::desc_Soluciones_By_Categoria_Equipo_Sintoma($sid_categoria_eq
uipoP,$sid_categoria_sintomaP,$sid_marcaP,$sid_modeloP,$sid_categoria_problemaP);

        $cont = 0;

        foreach ($data as $row)// por cada tupla, entro a comparar
cada atributo
        {
            $sid_caso = $row->id_caso;

            //dado el id de este caso, devuelveme todos los datos
de este caso
            $datos =
nomenclador_model::datos_dado_Id_Caso($sid_caso);

            $pesoTotal = 0;

            $pesoAtrTE = 0;

            $pesoAtrSI = 0;

            $pesoAtrMA = 0;

            $pesoAtrMO = 0;

            $pesoAtrCP = 0;

            $medio = 0.5;

            $bajo = 0.3;

            $alto = 0.7;

```

```

        foreach ($datos as $columna)
        {
            if ($id_tipo_equipoP==$columna->id_tipo_equipo)
            {
                $rasgo = "id_tipo_equipo";

                $resultPeso =
data_model::desc_Peso_dado_Atributo($rasgo);

                $pesoAtr = $resultPeso->desc_peso;

                $pesoAtrTE = $pesoAtr;

                //echo 'iguales y el peso es:
'. $pesoAtrTE. '<br>';
            }

            if ($id_sintomaP==$columna->id_sintoma)//sintoma
            {
                $rasgo = "id_sintoma";

                $resultPeso =
data_model::desc_Peso_dado_Atributo($rasgo);

                $pesoAtr = $resultPeso->desc_peso;

                $pesoAtrSI = $pesoAtr;

                //echo 'iguales y el peso es:
'. $pesoAtrSI. '<br>';
            }

            if ($id_marcaP==$columna->id_marca)//marca
            {
                $rasgo = "id_marca";

                $resultPeso =
data_model::desc_Peso_dado_Atributo($rasgo);

                $pesoAtr = $resultPeso->desc_peso;

                $pesoAtrMA = $pesoAtr;

                //echo 'iguales y el peso es:
'. $pesoAtrMA. '<br>';
            }

            if ($id_modeloP==$columna->id_modelo)//modelo
            {
                $rasgo = "id_modelo";

                $resultPeso =
data_model::desc_Peso_dado_Atributo($rasgo);

```



```

        $pesoAtr = $resultPeso->desc_peso;

        $pesoAtrMO = $pesoAtr;

        //echo 'iguales y el peso es:
'. $pesoAtrMO.' <br>';
    }

    if ($id_categoria_problemaP==$columna-
>id_categoria_problema)//categoria_problema
    {
        $rasgo = "id_modelo";

        $resultPeso =
data_model::desc_Peso_dado_Atributo($rasgo);

        $pesoAtr = $resultPeso->desc_peso;

        $pesoAtrCP = $pesoAtr;

        //echo 'iguales y el peso es:
'. $pesoAtrCP.' <br>';
    }
}

$pesoTotal = $pesoAtrTE + $pesoAtrSI + $pesoAtrMA
+ $pesoAtrMO + $pesoAtrCP;

    if ($pesoTotal >= $salto)//igual o mayor q
    {
        $cont++;
        $contRanking = $columna->ranking;
        $semejanza = $pesoTotal;
        $ouput .= '<tr align="center"
bgcolor="">
        <td><input onclick="" type="hidden"
id = "contador" name="contador" value="'. $cont.'" />
        <input onclick="verpasos(this);"
type="checkbox" id = "listarPasos" name="listarPasos" value="'. $row->id_caso.'"
/></td>
        <td>'. $row->desc_tipo_equipo.'</td>
        <td>'. $row->desc_sintoma.'</td>
        <td>'. $row->desc_solucion.'</td>
        <td
align="center">'. $contRanking.'</td>
        <td><strong>'. $semejanza.'</strong></td></tr>';
    }
    if (($pesoTotal >= $medio) && ($pesoTotal <
$salto))// menor e igual a 0.5 y 0.7
    {
        $cont++;
        $contRanking = $columna->ranking;

```

```

                $semejanza = $pesoTotal;
                $ouput .= '<tr align="center"
bgcolor=""><td><input onclick="" type="hidden" id = "contador" name="contador"
value="'. $cont.'" /><input onclick="verpasos(this);" type="checkbox" id =
"listarPasos" name="listarPasos" value="'. $row->id_caso.'" /></td><td>'. $row-
>desc_tipo_equipo.'</td><td>'. $row->desc_sintoma.'</td><td>'. $row-
>desc_solucion.'</td><td
align="center">'. $contRanking.'</td><td><strong>'. $semejanza.'</strong></td></tr>';
            }
            if (($pesoTotal >= $bajo) && ($pesoTotal <
$media))//entre 0.3 y 0.5
            {
                $cont++;
                $contRanking = $columna->ranking;
                $semejanza = $pesoTotal;
                $ouput .= '<tr align="center"
bgcolor=""><td><input onclick="" type="hidden" id = "contador" name="contador"
value="'. $cont.'" /><input onclick="verpasos(this);" type="checkbox" id =
"listarPasos" name="listarPasos" value="'. $row->id_caso.'" /></td><td>'. $row-
>desc_tipo_equipo.'</td><td>'. $row->desc_sintoma.'</td><td>'. $row-
>desc_solucion.'</td><td
align="center">'. $contRanking.'</td><td><strong>'. $semejanza.'</strong></td></tr>';
            }
        }
        $ouput .= '</tbody></table>';
        //pinto la tabla
        echo $ouput;

    }//fin del sintoma
    else // si no devuelve el conjunto de registros, ningun resultado
    {
        $ouput .= '</tbody></table>';
        //pinto la tabla
        echo $ouput;
    }
} //fin de ajax
} //fin de la funcion

```

3.2 Estándar de código.

Para el desarrollo de la aplicación se utilizó un estándar de código que permite el entendimiento de todo el código escrito. Los estándares de código son importantes y de obligatorio cumplimiento para los proyectos en la Universidad, ya que los equipos de desarrollo se desintegran con el tiempo y los sistemas deben ser mantenidos por equipos de soporte, que necesitan entender el código para posibles cambios. Se utilizarán los estándares definidos o internacionalmente aceptados.

Para PHP se utilizará la propuesta del grupo de desarrollo de la librería PEAR, según lo establecido en líneas bases de la arquitectura para el desarrollo de sistemas informáticos en la UCI, propuesta por la Dirección de Informatización.

3.3 Validación de la solución propuesta

Plantilla Caso de Prueba de aceptación. **Ver Anexo 4.**

La programación extrema define entre iteración e iteración un conjunto de casos de pruebas o test de aceptación para poder avanzar a una iteración superior. Durante el desarrollo de esta aplicación se realizaron un conjunto de pruebas a las que fue sometido el sistema para comprobar el funcionamiento de acuerdo a las Historias de Usuario. Se confeccionó un Plan de Prueba para organizar el desarrollo de las mismas

Se definieron casos de prueba para todas las historias de usuario, a continuación se dan a conocer las pruebas más significativas que se realizaron a las historias de usuario que se consideraron más importantes.

Caso de Prueba de Aceptación	
Código Caso de Prueba: Historia 1	Nombre Historia de Usuario: Autenticar usuario.
Nombre de la persona que realiza la prueba: Orlando Cruz-Claudio Puig	
Descripción de la Prueba: al intentar entrar al modulo de gestión del conocimiento el sistema debe pedir un usuario y contraseña. Se entrarán datos incorrectos para comprobar que no se conecten usuarios no deseados. Se entrarán los datos correctamente para comprobar que se conecten los usuarios autorizados.	
Condiciones de Ejecución: Que la PC tenga conexión de red para que pueda validar contra el dominio.	

<p>Entrada / Pasos de ejecución: En la ventana principal de la aplicación se selecciona el vínculo de Gestor de Conocimiento, el sistema pide el usuario y contraseña del dominio. El usuario teclea su usuario y contraseña y presiona Aceptar.</p>
<p>Resultado Esperado: Que no se pueda acceder al gestor del conocimiento si el usuario no está autorizado. En caso de estar autorizado que le permita entrar a este módulo.</p>
<p>Evaluación de la Prueba: Satisfactoria.</p>

Caso de Prueba de Aceptación	
<p>Código Caso de Prueba: Historia 3</p>	<p>Nombre Historia de Usuario: Mostrar casos dado un problema devuelto por el MI.</p>
<p>Nombre de la persona que realiza la prueba: Orlando Cruz-Claudio Puig.</p>	
<p>Descripción de la Prueba: se comprobara que al darle la descripción de un problema (equipo y síntoma) al sistema este devuelva los casos con mayor semejanza.</p>	
<p>Condiciones de Ejecución: Que estén definidos correctamente los rasgos objetivos para poder realizar la inferencia.</p>	
<p>Entrada / Pasos de ejecución: Después de estar descrito el problema pulsar el botón recuperar. El motor de inferencia a través de la fórmula de semejanza buscara los casos semejantes.</p>	
<p>Resultado Esperado: Que se listen los casos de mayor semejanza para el problema descrito.</p>	

Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación	
Código Caso de Prueba: Historia 6	Nombre Historia de Usuario: Adaptar solución a un nuevo problema.
Nombre de la persona que realiza la prueba: Orlando Cruz-Claudio Puig.	
Descripción de la Prueba: Dado un problema descrito por el usuario, el sistema propone una solución de otro caso con un problema similar, si la solución que este propone soluciona el problema antes descrito por el usuario se reorganiza la base de casos creando un nuevo problema, asignándole la solución propuesta.	
Condiciones de Ejecución: que exista un caso con un problema similar al que se presenta.	
Entrada / Pasos de ejecución: El sistema al realizar la inferencia debe mostrar los casos más semejantes. El usuario seleccionara el que considere correcto, seguirá los pasos y el sistema lo encuestara si resolvió con esa solución, en caso afirmativo se generará la adaptación.	
Resultado Esperado: Que adapte un caso que sea coherente.	
Evaluación de la Prueba: Satisfactoria	

3.4 Acerca de las funcionalidades obtenidas.

Teniendo en cuenta que el sistema se encuentra actualmente en su primera versión se pueden esperar muchos más resultados en posteriores versiones. Entre las principales funcionales que posee el SBCs actualmente se pueden mencionar:

- Se logró diseñar un modelo de datos capaz de almacenar los datos de los casos.
- La función de semejanza se implementó, el sistema devuelve los casos con rasgos semejantes ante un problema planteado.
- El sistema permite la adaptación de casos a partir de la reorganización de los mismos.
- Se obtuvo toda la documentación del sistema.

3.5 Conclusiones

Teniendo en cuenta las funcionalidades, requisitos no funcionales identificados y el modelo de diseño propuesto se llevó a cabo la implementación de la aplicación. Queda plasmado el código fuente del motor de inferencia, brindando una descripción del mismo. También se realizaron los casos de pruebas o test de aceptación propuestos por la metodología SXP, para comprobar el correcto funcionamiento del sistema de acuerdo a las HU.

Conclusiones Generales

- Se investigó acerca de la utilización de herramientas informáticas en la gestión del conocimiento, exactamente su incidencia en las TIC y beneficios en el proceso de gestión de incidencias y problemas tecnológicos.
- Se definieron las funcionalidades del sistema (20) a partir de la concepción inicial y estas quedaron recogidas en la LRP, siendo agrupadas en historias de usuario (18) para un mejor entendimiento por parte de los desarrolladores.
- Se implementó el sistema informático basado en casos para la gestión de incidentes tecnológicos en la Universidad de las Ciencias Informáticas utilizando la metodología ITIL, cumpliendo con las funcionalidades previstas para el mismo y teniendo en cuenta las normas técnicas y tecnologías aprobadas por la Dirección de Informatización de la UCI para el desarrollo de sistemas.

Recomendaciones

- Se continúe el desarrollo del presente trabajo con el fin de obtener nuevas funcionalidades, que aumenten la robustez, estabilidad del sistema propuesto.
- Liberar el sistema por el grupo de Calidad Central de la UCI.
- Extender la aplicación a otras instituciones que brinden servicios de soporte técnico.
- Integración del sistema con los principales servicios que se ofrecen en la intranet universitaria.

Referencias bibliográficas

- [1] Sociedad Española de Documentación e Información científica. (2000). 2 p. *La Gestión del Conocimiento:Contexto,concepto y aplicaciones*. Disponible en: <http://www.sedic.es/emoreno.pdf>.
- [2] Sociedad Española de Documentación e Información científica. (2000). 9 p. *La Gestión del Conocimiento:Contexto,concepto y aplicaciones*. Disponible en: <http://www.sedic.es/emoreno.pdf>.
- [3] *Gestión del conocimiento II*. (18 de 08 de 2008). Disponible en: http://www.egestionconsultoria.com/web/index.php?option=com_content&view=article&id=15:boletin2&catid=7:boletines&Itemid=12.
- [4] Nieto, Teresa Lucio (n.d.). *Las Herramientas para la Administracion del conocimiento(KM)*.
- [5] Univesidad Nacional de Colombia. *Sistemas basados en conocimiento*. Disponible en dis.unal.edu.co/profesores/lucas/iartificial/IA00111.pdf.
- [6] Idem [5]
- [7] Lio, Dr. Daniel Gálvez (1998). Cap 2, 3 p. *Sistemas Basados en el Conocimiento*. Universidad Central "Martha Abreu", Villa Clara.
- [8] Lio, Dr. Daniel Gálvez (1998). Cap 6, 4 p. *Sistemas Basados en el Conocimiento*. Universidad Central "Martha Abreu", Villa Clara.
- [9] Lio, Dr. Daniel Gálvez (1998). Cap 8, 4 p. *Sistemas Basados en el Conocimiento*. Universidad Central "Martha Abreu", Villa Clara.
- [10] Edwin Andres Bernal Lopez. Universidad Nacional de Colombia. *Sistema Prototipo Basado en el Conocimiento*. Disponible en: <http://dis.unal.edu.co/profesores/ypinzon/2013326-206/docs/Tesis1Bernal.pdf>
- [11] César A. Chagoyén Méndez.Centro de Estudios CAD/CAM, F. d. Ingeniería Mecánica, UCLV (1999-2000). 64 p. *Un sistema basado en casos para diseño de árboles*. Disponible en: <http://biblioteca.idict.villaclara.cu/UserFiles/File/ciencia/72.pdf>.

[12] Ochoa A. Hernández, Universidad Autónoma de Aguascalientes Editorial (2006). *Más allá del razonamiento basado en casos y una aproximación al modelado de sociedades utilizando minería de datos*. Disponible en: <http://www.uaem.mx/cicos/memorias/5tocic2006/Articulos/articulo11.pdf>.

[13] César A. Chagoyén Méndez. Centro de Estudios CAD/CAM, F. d. Ingeniería Mecánica, UCLV (1999-2000). 67 p. *Un sistema basado en casos para diseño de árboles*. Disponible en: <http://biblioteca.idict.villaclara.cu/UserFiles/File/ciencia/72.pdf>.

[14] Lio, Dr. Daniel Gálvez (1998). Cap 6, 2 p. *Sistemas Basados en Casos*. Universidad Central "Martha Abreu", Villa Clara.

[15] Colectivo de Autores. Ken Schwaber, Jeff Sutherland, Dave Thomas. *Manifesto for Agile Software Development*. Disponible en: <http://www.agilealliance.org>.

Bibliografía

1. Sociedad Española de Documentación e Información científica. (2000). *La Gestión del Conocimiento: Contexto, concepto y aplicaciones*. Disponible en: <http://www.sedic.es/emoreno.pdf>.
2. *Gestión del conocimiento II*. (18 de 08 de 2008). Obtenido de http://www.egestionconsultoria.com/web/index.php?option=com_content&view=article&id=15:boletin2&catid=7:boletines&Itemid=12.
3. Nieto, Teresa Lucio (n.d.). *Las Herramientas para la Administración del conocimiento (KM)*.
4. Univesidad Nacional de Colombia. *Sistemas basados en conocimiento*. Disponible en dis.unal.edu.co/profesores/lucas/iartificial/IA00111.pdf.
5. Lio, Dr. Daniel Gálvez (1998). *Sistemas Basados en el Conocimiento*. Universidad Central "Martha Abreu", Villa Clara.
6. Lio, Dr. Daniel Gálvez (1998). *Sistemas Basados en Casos*. Universidad Central "Martha Abreu", Villa Clara.
7. Lopez, Edwin Andres Bernal. Universidad Nacional de Colombia. *Sistema Prototipo Basado en el Conocimiento*. Disponible en: <http://dis.unal.edu.co/profesores/ypinzon/2013326-206/docs/Tesis1Bernal.pdf>.
8. Méndez, César A. Chagoyén. Centro de Estudios CAD/CAM, F. d. Ingeniería Mecánica, UCLV (1999-2000). *Un sistema basado en casos para diseño de árboles*. Disponible en: <http://biblioteca.idict.villaclara.cu/UserFiles/File/ciencia/72.pdf>.
9. Ochoa A. Hernández, Universidad Autónoma de Aguascalientes Editorial (2006). *Más allá del razonamiento basado en casos y una aproximación al modelado de sociedades utilizando minería de datos*. Disponible en: <http://www.uaem.mx/cicos/memorias/5tocic2006/Articulos/articulo11.pdf>.
10. Dirección de Informatización. (05 de 2007). *Arquitectura para los Sistemas que Conforman la Intranet Universitaria*. Disponible en: <http://uddi.uci.cu/files/arquitectura.2007.5.9.pdf>.

11. Colectivo de Autores. Ken Schwaber, Jeff Sutherland, Dave Thomas. *Manifiesto for Agile Software Development*. Disponible en: <http://www.agilealliance.org>.
12. Villar, Malay Rodríguez. (2007). *Introducción de procedimientos ágiles en la producción de software en la Facultad 7 de la Universidad de las Ciencias Informáticas*. Disponible en: http://bibliodoc.uci.cu/TD/TD_0693_07.pdf.
13. Romero, Gladys M. Peñalver. (2008). *MA-GMPUR2 Metodología ágil para proyectos de software libre*. Universidad de las Ciencias Informáticas, Facultad 10. Disponible en: http://bibliodoc.uci.cu/TD/TD_0693_07.pdf.
14. Gutiérrez, J. (2007). *Metodologías Ágiles*. Universidad Pablo de Olavide. Kniberg, Henrik. *Scrum y XP desde las trincheras*. Estados Unidos. C4Media, 2007. ISBN: 978-1-4303-2264-1.
15. Matos, Jorge L. (2008). *Sistema de Gestión de Tecnología de la Información de la Universidad de las Ciencias Informáticas: Diseño e implementación del módulo Solicitud de Servicios*. Disponible en: http://bibliodoc.uci.cu/TD/TD_1734_08.pdf.
16. Roldán, Dayron Pérez. (2008). *Sistema de Clonación y Distribución de Imágenes de Sistemas Operativos*. Disponible en: http://bibliodoc.uci.cu/TD/TD_1245_08.pdf.
17. Schuller, Joseph. (2000). *Aprendiendo UML en 24 Horas*. México: Pearson Educación.
18. Céspedes, R. F., & García, S. P. (2008). *Propuesta de un expediente, para los proyectos productivos del Polo de Software Libre, de la Facultad 10*. Disponible en: http://bibliodoc.uci.cu/TD/TD_1247_08.pdf.
19. *DICCIONARIO DE LA LENGUA ESPAÑOLA* . Obtenido de <http://www.rae.es/rae.html>
20. Conocimiento Virtual Academia, Colombia. *Code Igniter en Español*. Conocimiento Virtual Academia Ltda.
21. *Pear*. Recuperado el 20 de mayo de 2009, de <http://pear.php.net/manual/en/standards.php>.

Anexos

Anexo 1. Tipos de Sistemas Basado en el Conocimiento.

Nombre	Forma de Representación del Conocimiento (FRC)	Explicación	Método de solución del Problema (MSP)	Fuentes de Conocimiento
Sistemas basados en reglas (SBR)	Reglas de producción	Reglas activas	Usualmente búsqueda primero en profundidad con dirección backward o forward	Expertos, publicaciones, ejemplos
Sistemas basados en frames (SBF)	Frames		Herencia y procedimientos adjuntos	Expertos, publicaciones, ejemplos
Sistemas basados en casos (SBCSasos)	Casos	Casos semejantes	Razonamiento basado en casos (búsqueda por semejanza y adaptación de las soluciones)	Ejemplos
Sistemas basados en probabilidades (SBP)	Probabilidades o frecuencias	Valores de probabilidades condicionales	Teorema de Bayes y otras técnicas de inferencia estadística	Ejemplos
Redes expertas	Pesos y alguna otra FRC	Según el modelo simbólico	Cálculo de niveles de activación de las neuronas	Ejemplos

Sistemas basados en modelos	Modelo del artefacto		Razonamiento basado en modelos	Esquemas estructurales y funcionales del artefacto
-----------------------------	----------------------	--	--------------------------------	--

Anexo 2. Guión de la Metodología SXP.

GUIÓN DE LA METODOLOGÍA

Planificación ↔ Definición

- Entrevista con el cliente (concepción inicial)
 - **Plantilla de concepción del sistema.(Actualizó)**
- Juego de la planificación.
 - **Plantilla Modelo de Historia de Usuario del negocio.**
- Captura de requisitos:
 - Creación de la LRP.
 - **Plantilla LRP (Lista de Reserva del Producto).**
 - Priorización de la LRP.
 - Definir las historias de usuario.
 - **Plantilla Historia de usuario.**
 - Asignar las historias de usuario.
 - **Plantilla Historia de usuario. (actualizar gerente.)**
- Valoración del esfuerzo.
 - **Plantilla Historia de usuario. (actualizar gerente.)**
- Valoración de riesgos.
 - **Plantilla Lista de riesgos.**
- Diseño con las metáforas.
 - **Plantilla Modelo de diseño.**
- Refactorización.
- Reunión de revisión del diseño.

Desarrollo

- Junta de planificación.
 - **Plantilla de Glosario de términos.**
 - Definir las historias de usuario a implementar.

- Tareas para lograr dicha implementación.
 - Plantilla de Tareas de Ingeniería.
- Plantilla Cronograma de producción.
- Plantilla de Plan de releases.
- Implementación.
 - Estándar de código.
 - Código fuente.
- Junta de seguimiento
- Taller técnico.
- Junta de revisión.
- Pruebas
 - Plan de Pruebas.

 - Plantilla Caso de Prueba de aceptación.

Entrega

- Entrega de la documentación.
- Entrenamiento.
 - Capacitación
 - Manual de usuario. (Programadores)
 - Manual de Identidad (Diseñadores)
 - Manual de desarrollo. (Desarrollador, y se genera automáticamente.)
 - Instalación.
- Marketing

Mantenimiento

- Soporte.
 - Plantilla de Gestión de cambios.

Observación: Desde el diseño en la fase de Planificación ↔ Definición y toda la fase de Desarrollo es un ciclo.

Anexo 3. Plantilla Historia de Usuario.

Historia de Usuario	
Número: <i>[Número de la historia]</i>	Nombre Historia de Usuario: <i>[Nombre que identifica la historia.]</i>
Modificación de Historia de Usuario Número: <i>[Cantidad de modificaciones que se le ha realizado a la historia de usuario (de no tener modificaciones se pone ninguna, sino la cantidad de veces que ha sido modificada).]</i>	
Usuario: <i>[Programador responsable de su implementación]</i>	Iteración Asignada: <i>[Que iteración se desarrollará. (Según su importancia)]</i>
Prioridad en Negocio: <i>[Prioridad puede ser Alta, Media o Baja(Según Cliente)]</i>	Puntos Estimados: <i>[Tiempo en semanas que se le asignará. (Estimado)]</i>
Riesgo en Desarrollo: <i>[Riesgo puede ser Alto, Medio o Bajo (Según Programadores)]</i>	Puntos Reales: <i>[Tiempo real dedicado a la realización de la HU en semanas.]</i>
Descripción: <i>[Breve descripción del proceso que define la historia.]</i>	
Observaciones: <i>[Alguna acotación importante de señalar acerca de la historia.]</i>	
Prototipo de interfase: <i>[Imagen de cada una de las interfaces relacionadas con la HU.]</i>	

Anexo 4. Plantilla Caso de Prueba de Aceptación.

Caso de Prueba de Aceptación	
Código Caso de Prueba: <i>[Inicial del proyecto-número de la HU a la que pertenece la prueba-número de la prueba.]</i>	Nombre Historia de Usuario: <i>[Nombre de la HU a realizar prueba.]</i>
Nombre de la persona que realiza la prueba: <i>[Nombre y apellidos.]</i>	
Descripción de la Prueba: <i>[Descripción de la prueba realizada.]</i>	
Condiciones de Ejecución: <i>[Condiciones necesarias para poder realizar la prueba.]</i>	
Entrada / Pasos de ejecución: <i>[Serie de pasos necesarios para lograr la realización de la HU, y así realizar la prueba.]</i>	
Resultado Esperado: <i>[Que cumpla con las restricciones del producto.]</i>	
Evaluación de la Prueba: <i>[Satisfactoria o no satisfactoria.]</i>	

Glosario de Términos

A

Applet: es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web.

Ajax: acrónimo de **A**synchronous **J**avaScript **A**nd **X**ML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o **RIA** (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

C

Caso: es un conjunto arbitrario de rasgos usado para describir un concepto particular. Consiste en la descripción de un problema y la solución dada.

Checkbox: En informática, un checkbox es un elemento de la interfaz gráfica de usuario que permite al usuario hacer selecciones múltiples de un conjunto de opciones.

E

Experto: es una persona reconocida como una fuente confiable de un tema, técnica o habilidad cuya capacidad para juzgar o decidir en forma correcta, justa, o inteligente le confiere autoridad y estatus por sus pares o el público en un dominio específico.

Equipo: En informática, al conjunto de aparatos y dispositivos que constituyen el material de un ordenador.

F

Framework: en el desarrollo de software, es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

G

Gestionar el conocimiento: es un concepto aplicado en las organizaciones, que pretende transferir el conocimiento y experiencia existente entre sus miembros, de modo que pueda ser utilizado como un recurso disponible para otros en la organización.

H

Heurística: se denomina heurística a la capacidad de un sistema para realizar de forma inmediata innovaciones positivas para sus fines. La capacidad heurística es un rasgo característico de los humanos. Trata de métodos exploratorios durante la resolución de problemas en los cuales las soluciones se descubren por la evaluación del progreso logrado en la búsqueda de un resultado final.

Herramienta CASE: Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

Historia de usuario: término en el cual se resume todo lo referente a las especificaciones de los requisitos, así como lo relacionado con los casos de uso, conocidos del proceso unificado.

I

Incidente: Hecho que interrumpe u obstaculiza un acto o asunto.

Incidentes tecnológicos: rotura de un equipo o interrupción de un servicio.

Inteligencia Artificial (IA): rama de la ciencia informática. Aplica o involucrada a términos como Robótica, Autómatas, Sistemas Expertos, es una disciplina que envuelve a varias ramas de estudio: la ingeniería, la computación, la psicología, la física, la medicina, la filosofía, la teología y lo que se acumule.

ITIL: siglas del inglés, Information Technologies Infraestructura Library. Biblioteca de Infraestructura de Tecnologías de Información.

Inferencia: es una evaluación que realiza la mente entre conceptos que, al interactuar, muestran sus propiedades de forma discreta, necesitando utilizar la abstracción para lograr entender las unidades que componen el problema, creando un punto, que nos permitirá trazar una línea lógica de causa-efecto, entre los diferentes puntos inferidos en la resolución del problema.

Iteraciones: En el contexto de un proyecto se refieren a la técnica de desarrollar y entregar componentes incrementales de funcionalidades de un negocio. Una iteración resulta en uno o más paquetes atómicos y completos del trabajo del proyecto que pueda realizar alguna función tangible del negocio. Múltiples iteraciones contribuyen a crear un producto completamente integrado.

Infraestructura tecnológica: equipamiento tecnológico.

M

Modus Ponens: En latín, "modo que afirma".

Máquina virtual Java: una Máquina virtual Java (en inglés *Java Virtual Machine*, JVM) es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java.

Marca: Una Marca es un título que concede el derecho exclusivo a la utilización de un signo para la identificación de un producto o un servicio en el mercado.

Modelo: Objeto, aparato, construcción o conjunto de ellos realizados con arreglo a un mismo diseño.

P

Primera línea de soporte: Es el primer contacto con el usuario, se implementa en centros de llamada o mesas de ayuda, su objetivo es resolver en el menor tiempo las interrupciones del servicio. En esta línea si se resuelve el incidente se cierra la orden, si no se escala a la segunda línea.

R

Reporte: Referente a cuando el usuario anuncia una rotura.

Ranking: Escala de importancia.

Razonamiento: El término razonamiento se define de diferente manera según el contexto, normalmente se refiere a un conjunto de actividades mentales consistentes en conectar unas ideas con otras de acuerdo a ciertas reglas o también puede referirse al estudio de ese proceso. Se entiende por razonamiento la facultad humana que permite resolver problemas.

Release: Entrega.

S

Sistemas Expertos: son llamados así porque emulan el comportamiento de un experto en un dominio concreto y en ocasiones son usados por estos. Con los sistemas expertos se busca una mejor calidad y rapidez en las respuestas dando así lugar a una mejora de la productividad del experto.

Sprint: la traducción al español es carrera breve a todo correr, correr velozmente.

Un Sprint es el procedimiento de adaptación de las cambiantes variables del entorno (requisitos, tiempo, recursos, conocimiento, tecnología). Son ciclos iterativos en los cuales se desarrolla o mejora una funcionalidad para producir nuevos incrementos. Durante un Sprint el producto es diseñado, codificado y probado. Y su arquitectura y diseño evolucionan durante el desarrollo.

Segunda línea de ayuda: Conformada por los especialistas.

T

Tecnología de la Información (TI): se entiende como las herramientas y métodos empleados para recabar, retener, manipular o distribuir información. La tecnología de la información se encuentra generalmente asociada con las computadoras y las tecnologías afines aplicadas a la toma de decisiones.

U

UML: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

W

Workarounds: soluciones alternativas, ajustes. Implica una solución que logra resolver el problema de una manera indirecta, ejemplo, evitarla de alguna forma, en vez de enfrentarla directamente.