

Universidad de las Ciencias Informáticas

Facultad 1

Título: Interfaces para el módulo de Administración del Sistema de Gestión de Servicios de Soporte del Centro de Soporte UCI

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

Autores: Macdemis Liliana Pozo Guevara

Tutores: MSc. Yulio Seriocha García

Ing. Neybis Lago Clara

Co-tutor: Ing. Juan Manuel Álvarez Tur

“La Habana, Junio del 2014”



“Los seres humanos sin conocimientos, se asemejan a un meteoro viajando sin rumbo en el espacio sideral.”

Sidbarto

DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 18 días del mes de junio del año 2014.

Macdemis L. Pozo Guevara
Firma del Autor

MSc. Yulio Seriocha García
Firma del Tutor

Ing. Neybis Lago Clara
Firma del Tutor

Ing. Juan Manuel Álvarez Tur.
Firma del Co - Tutor

DATOS DE CONTACTO

Autor: Macdemis Liliana Pozo Guevara.

Correo: mlpozo@estudiantes.uci.cu

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Tutor: Ing. Neybis Lago Clara.

Correo: nlago@uci.cu

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Tutor: MSc. Yulio Seriocha García Gallardo.

Correo: ygarciag@uci.cu

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Co-Tutor: Ing. Juan Manuel Álvarez Tur.

Correo: jmalvarez@uci.cu

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Dedicatoria

Dedico esta tesis a mi linda mamita y a mi padre querido por haberme apoyado y ayudado a realizar todas las metas que me he trazado. Ellos han sido un gran ejemplo para mí y me siento dichosa por tener unos padres tan maravillosos como ellos. En este pequeño espacio les ofrezco mi mayor regalo y quiero que sepan que nunca los defraudaré. No existen personas más importantes en mi vida que ustedes dos, por tal motivo mi trabajo de diploma para optar por el título de Ingeniera en Ciencias Informáticas va dedicado a ustedes.

Macdemis Liliana Pozo Guevara.

Agradecimientos

A mi preciosa madre y a mi lindo padre, gracias por creer y confiar en su única hija, los amo.

A los mejores amigos que todo el mundo quisiera tener, los muchachos del 143 201, Yoandri, Dennis, Raúl y Dayrol.

A Yoandri por poner a mi disposición su laptop y parte de su tiempo, gracias amigo, tu ayuda fue incondicional.

Al chino, el angel que apareció cuando pensé que mi último año sería el más triste, gracias por tu paciencia, tu comprensión, tu apoyo y por toda la buena vibra que transmites.

A Roxana por escucharme cuando lo necesitaba y por sacarme de varios apuros.

A una madre lindísima por dentro y por fuera, Violeta, gracias por todas las buenas acciones que has hecho por mí.

A una china linda que apoyó mucho a mi mamá en Santiago y que es madre del hombre que amé, Olga, gracias.

A mis tutores por el apoyo para el desarrollo de este trabajo.

A todas aquellas personas que de una manera u otra han sido partícipes de mi formación durante estos largos años de estudio.

Macdemis Liliana Pozo Guevara.

Resumen

La Gestión de Servicios de Tecnologías de la Información (GSTI), se ha convertido en la clave principal para lograr el aumento de la calidad de los servicios ofrecidos por empresas proveedoras de TI (Tecnologías de la Información). Por lo que para poder realizar una correcta gestión del servicio a lo largo de todo su ciclo de vida, es necesario adoptar el estándar adecuado. Dentro de estos se encuentran ISO/IEC 20000, CMMI SVC, COBIT e ITIL.

El presente trabajo tiene como objetivo desarrollar las Interfaces de un Módulo de Administración basadas en el estándar ITIL. Este módulo debe ser capaz de permitir la futura integración de los módulos de Gestión de Incidencias, Problemas, Cambios, Disponibilidad, Catálogo, Niveles de Servicios y la Base de Datos de la Gestión de Configuraciones (*CMDB, por sus siglas en inglés*); así como también la posterior gestión de los procesos involucrados en el Sistema de Gestión de Servicios del Centro de Soporte de la Universidad de las Ciencias Informáticas (UCI). Contribuyendo de esta forma a mejorar la calidad de los servicios que se prestan en el centro.

Para el desarrollo del mismo se utilizaron las herramientas que fueron asignadas por políticas del Centro de Soporte. Como lenguaje de programación: Groovy, como framework de desarrollo Grails 2.1.1 y como servidor web: Apache Tomcat 2.1.1. El sistema de gestión de bases de datos a utilizar: PostgreSQL 9.2. Utilizando además la biblioteca Bootstrap 2.2.1 y para el diseño y documentación de las interfaces del módulo, se utilizará IntelliJIDEA 12.2.3 y la metodología de desarrollo XP. Culminando la validación del sistema mediante las pruebas unitarias, de aceptación y las de carga y estrés.

Palabras claves: Gestión de Servicios de Tecnologías de la Información, Interfaces del Módulo de Administración, Tecnologías de la Información.

ÍNDICE

Introducción 1

Capítulo 1: FUNDAMENTACIÓN TEÓRICA DE LAS INTERFACES PARA EL MÓDULO DE ADMINISTRACIÓN DEL SISTEMA DE GESTIÓN DE SERVICIOS DE SOPORTE 6

1.1 Introducción..... 6

1.2 Conceptos Fundamentales..... 6

1.3 Service Desk 7

1.4 Estándares para la gestión de servicios TI 11

1.4.1 COBIT 11

1.4.2 ISO/IEC 20000 12

1.4.3 CMMI-SVC 12

1.4.4 ITIL..... 13

1.5 ITIL..... 14

1.5.1 Metodología que propone ITIL para brindar servicios de soporte 17

1.6.1 NetSupport ServiceDesk 21

1.6.2 CA Service Desk Manager 21

1.6.3 Servicio de Referencia Virtual y Call Center. 22

1.6.4 ManageEngine Service Desk Plus 22

1.7 Metodologías de Desarrollo..... 24

1.8 Herramientas y tecnologías 24

1.8.1 IntelliJIDEA 12.2.3..... 25

1.8.2 Apache Tomcat 2.1.1 25

1.8.3 Groovy 25

1.8.4 Grails 2.1.1 25

1.8.5 PostgreSQL 9.2..... 26

1.8.6 BootStrap 2.2.1	26
1.8.7 Visual Paradigm 5	26
1.8.8 JMeter	26
1.9 Conclusiones Parciales.	26
Capítulo 2: Propuesta del Sistema	28
2.1 Introducción.....	28
2.2 Módulo de Administración para el Sistema de Gestión de Servicios de Soporte de la UCI.....	28
2.3 Requisitos del sistema	28
2.4 Fase de Exploración.....	33
2.4.1 Roles y Responsabilidades	33
2.4.2 Historias de usuario.....	34
2.5 Fase de planificación.....	36
2.5.1 Estimaciones del esfuerzo.....	36
2.5.2 Plan de duración de iteraciones	37
2.5.3 Plan de entrega.....	38
2.6 Diseño del sistema	39
2.6.1 Tarjetas CRC	39
2.6.2 Patrones de diseño	41
2.6.3 Patrones de arquitectura	44
2.7 Modelo Conceptual	47
2.7.1 Descripción del modelo conceptual	49
2.8 Diseño de la base de datos	50
2.9 Fase de implementación	52
2.9.1 Estándares de codificación.....	52
2.9.2 Implementación.....	53

2.10 Conclusiones parciales.....	57
Capítulo 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	58
3.1 Introducción.....	58
3.2 Pruebas.....	58
3.2.1 Pruebas unitarias	58
3.2.2 Pruebas de aceptación.....	59
3.2.3 Pruebas de Carga	60
3.3 Conclusiones parciales	62
Conclusiones generales.....	63
Recomendaciones	64
Bibliografía	68
Glosario de Términos.....	70
Anexos.....	71

ÍNDICE DE FIGURAS

Figura 1. Tipos de Service Desk según su estructura física.	11
Figura 2. Enfoque Estratégico de ITIL.	16
Figura 3. Procesos oficiales de ITIL en correspondencia con sus fases. (Corona, 2011)	17
Figura 4. Principales aspectos de la metodología de soporte al servicio. (OSIATIS, 2010)	20
Figura 5. Integración de los módulos.....	28
Figura 6. Ejemplo del uso del patrón Experto.....	42
Figura 7. Ejemplo del uso del patrón creador.....	43
Figura 8. Patrón Modelo Vista Controlador.....	45
Figura 9. Ejemplo del componente modelo en el sistema.....	45
Figura 10. Ejemplo del componente vista en el sistema.	46

Figura 11. Ejemplo del componente controlador en el sistema.....	47
Figura 12. Modelo Conceptual.	49
Figura 14. Modelo Físico.....	51
Figura 15. Prueba unitaria realizada al método List de la clase TareaController.	58
Figura 16. Parámetros de rendimiento evaluados por la herramienta JMeter.....	61
Figura 17. Prueba Unitaria realizada al método Create de la clase TareaController.....	78
Figura 18. Prueba Unitaria realizada al método save de la clase TareaController.....	78
Figura 19. Vista de la interfaz del Módulo de Administración	79
Figura 20. Vista de la interfaz General del Sistema de Gestión de Servicios de Soporte	80

ÍNDICE DE TABLAS

Tabla 1. Procesos de las áreas Soporte de Servicio y Prestación de Servicio.	18
Tabla 2. Lista de reserva del producto. Requisitos funcionales.	29
Tabla 3. Lista de reserva del producto. Requisitos no funcionales.	33
Tabla 4. Historia de Usuario 1	35
Tabla 5. Historia de Usuario 2.....	36
Tabla 6. Plan de esfuerzo por Historia.	37
Tabla 7. Plan de duración de las iteraciones.....	38
Tabla 8. Plan de entrega de las iteraciones.....	39
Tabla 9. Tarjeta CRC Comentario.	40
Tabla 10. Tarjeta CRC Tareas.	40
Tabla 11. HU abordadas en la iteración I	53
Tabla 12. Descripción de la tarea de desarrollo #2.....	54
Tabla 13. HU abordadas en la iteración 2	55
Tabla 14. Descripción de la tarea de desarrollo #1.....	55
Tabla 15. HU abordadas en la iteración 3	56
Tabla 16. Descripción de la tarea de desarrollo #8.....	56
Tabla 17. Prueba de aceptación de la historia de usuario 1	60
Tabla 18. Historia de Usuario 2.1	71

Tabla de contenido

Tabla 19. Historia de Usuario 2.2	72
Tabla 20. Tarjeta CRC Aviso.....	73
Tabla 21. Tarjeta CRC Permiso	73
Tabla 22. Tarjeta CRC Rol	74
Tabla 23. Prueba de Aceptación de la Historia de Usuario 2.2.....	75
Tabla 24. Prueba de Aceptación de la Historia de Usuario 2.3.....	76
Tabla 25. Prueba de Aceptación de la Historia de Usuario 2.4.....	77

INTRODUCCIÓN

A lo largo de los años estar a la par con los avances tecnológicos se ha vuelto una cuestión indispensable. Por esta razón, las Tecnologías de la Información juegan un papel relevante en el éxito y desarrollo de cada empresa, pues para muchas, la correcta gestión de la información y la tecnología relacionada que la soporta, representan sus activos más valiosos.

“Estas empresas en la actualidad se enfrentan a una enorme presión dirigida hacia la reducción de costes, a una fortísima competencia, a unos recursos cada vez más escasos y a continuos cambios en la manera de operar” (Sun Microsystems, 2003). Todas estas razones aumentan la dependencia del uso de las TI para así poder cumplir con sus metas y objetivos organizacionales; así como también para alcanzar satisfactorios resultados en el negocio, proporcionándoles a los clientes resultados óptimos. Por lo que la mejor manera de mantener estas características de empresa emprendedora, es la utilización del uso de las tecnologías de la información.

Sin embargo, son muchas las preocupaciones existentes en las empresas proveedoras de TI, en cuanto a los servicios que ofrecen, debido a que estos muchas veces no se ajustan a sus necesidades de negocios y a las exigencias de sus clientes; creciendo al mismo tiempo sus expectativas por la innovación, la mejora de la calidad y los resultados tangibles que pueden ser alcanzados; *“pues entregar un servicio constante, fiable y estable que agregue valor al negocio, se ha convertido en un reto para estas entidades, dado que los clientes compran satisfacción, no sólo productos o servicios” (Vilches, 2010).* Siendo imprescindible que tomen un enfoque centrado principalmente en el cliente, para así poder concentrarse en la calidad de los servicios que brindan y asegurarse de que estos se ajusten a los objetivos de la organización.

Todo lo anteriormente mencionado demuestra que la utilización solamente de las TI, no cubre las necesidades y expectativas de las organizaciones y sus clientes. A raíz de esto se hace necesario y de vital importancia, la introducción y el uso de un esencial concepto como La Gestión de Servicios de Tecnologías de la Información GSTI (*en inglés IT Service Management, ITSM*) la cual es conceptualizada como una *“disciplina basada en procesos, enfocada en alinear los servicios de TI proporcionados con las necesidades de las empresas, poniendo énfasis en los beneficios que puede percibir el cliente final” (Ingenius, 2009).*

Aunque la gestión de servicios de TI es una buena alternativa para cubrir las expectativas y necesidades de las organizaciones y sus clientes, son muchos los problemas que surgen a la hora de gestionar estas

tecnologías. “Dentro de los más importantes se encuentran la mala gestión de los proyectos, gestión inadecuada de servicios, falta de alineamiento estratégico para las iniciativas, métodos de desarrollo de software inadecuados, así como las inversiones en TI que no generan inversiones tangibles” (Coello, 2008).

Con el fin de erradicar estos problemas se han creado diversos marcos de trabajo y mejores prácticas. “*Estos estándares son la base para lograr un buen Gobierno de TI, que permitirá asegurar la obtención de beneficios de las inversiones realizadas, en base a la correcta gestión de riesgos, recursos y alineamiento estratégico*” (Coello, 2008). Dentro de estos se encuentran: COBIT¹, ISO/IEC 20000², CMMI-SVC³ e ITIL. Es importante destacar que para que estas mejores prácticas resulten efectivas, no deben ser tratadas como puras guías técnicas (Instituto de Gobierno de TI, 2008). Cada organización debe saber aplicar el modelo que mejor se adapte a sus objetivos y necesidades, pues son muchos los que existen hoy en día.

COBIT es un marco de referencia globalmente aceptado para el gobierno de TI basado en estándares de la industria y buenas prácticas (Instituto de Gobierno de TI, 2008). Este ha demostrado ser útil para las organizaciones que pretenden mejorar el monitoreo y control del uso de las mismas, pues sus herramientas así lo permiten.

La ISO/IEC 20000 es el primer estándar internacional exclusivo para el área de gestión de servicios TI (Layme Velázquez, Rubén Darío, 2011). Esta proporciona la base para probar que una organización de TI ha implantado buenas prácticas para la gestión del servicio y que las está usando de forma regular y consistente.

El modelo de CMMI-SVC es un modelo de capacidad que ofrece el marco de referencia adecuado para administrar y entregar los servicios ofertados, proporcionando una guía para aplicar buenas prácticas en una organización que provee servicios (Avantare, 2014).

Biblioteca de Infraestructura de Tecnologías de Información (*ITIL, por sus siglas en inglés*), la cual puede ser definida como “*un conjunto de buenas prácticas destinadas a mejorar la gestión y provisión de servicios TI*” (Osatis, 2010). Encargándose además de proporcionar una serie de procesos para entregar con alta calidad la provisión y el soporte de los mismos.

¹ COBIT: Objetivos de Control para la Información y Tecnologías.

² ISO/IEC 20000: Organización Internacional para Estandarización/Comisión Electrónica Internacional.

³ CMMI-SVC: Modelo de Madurez de la Capacidad Integrado para Servicios.

La gestión de servicios tiene su columna vertebral en la función de Centro de Servicios (*en inglés Service Desk*), concepto que plantea ser el único punto de entrada y salida para prestar servicios de soporte. En otras palabras es el punto de contacto entre los usuarios necesitados y los técnicos encargados de brindar soporte a herramientas y aplicaciones (Coello, 2008), por lo que se puede afirmar que es una potente herramienta web que ha surgido como una alternativa para mejorar la calidad de los servicios de soporte técnico, facilitándoles el trabajo a los usuarios.

Cuba con el propósito de elevar el nivel cultural de la población y su calidad de vida, ha realizado novedosas transformaciones educacionales y sociales como parte de los programas de la Batalla de Ideas dirigidos por Fidel Castro Ruz. Una de ellas fue la creación en el año 2002 de La Universidad de las Ciencias Informáticas (UCI), la cual constituye un pilar en el desarrollo de la Industria Cubana del Software y en la materialización de los proyectos asociados al programa cubano de informatización de la sociedad.

La UCI se compone de varios centros que se especializan en el desarrollo de aplicaciones informáticas, dentro de estos se encuentra el Centro de Soporte UCI. Este centro es el encargado de prestar servicios de soporte técnico a los usuarios en las aplicaciones desarrolladas por cada centro productivo de la Universidad. Para gestionar la prestación de los servicios TI, el centro cuenta con una plataforma. Aun así el trabajo del equipo de soporte técnico se ve afectado en su desempeño por la falta de calidad en algunos factores específicos.

Después de analizar el trabajo realizado en el centro, se pudo comprobar que la prestación de los servicios se ve afectada. Siendo uno de los problemas más importantes, la carencia de una herramienta capaz de gestionar todos los procesos involucrados en el Sistema de Gestión de Servicios de Soporte, que actualmente se desarrolla en dicho centro. Dicha herramienta debe permitir una futura integración de los módulos de Gestión de Incidencias, Problemas, Cambios, Disponibilidad, Catálogo, Niveles de Servicios y la Base de Datos de Gestión de Configuraciones (*CMDB, por sus siglas en inglés*).

La plataforma utilizada para la prestación de estos servicios es de carácter propietaria y no se gestionan correctamente los eventos que se producen cuando un servicio o dispositivo alcanza un umbral, se ha producido un fallo o simplemente ha cambiado algo y puede llegar a convertirse en una incidencia. Tampoco se cuenta con una base de datos de eventos; se incumple en los tiempos de respuesta y resolución establecidos con el cliente; existen demoras por parte de los especialistas en determinar la causa raíz y los síntomas ante un problema detectado y por último se desconocen los cambios realizados a las aplicaciones informáticas, lo que provoca demoras en el servicio a los clientes. Todo esto se debe a

la falta de comunicación entre los procesos que intervienen en la gestión de servicios TI, influyendo en su calidad de manera directa.

Por todo lo anteriormente expuesto se plantea el siguiente **problema a resolver**: ¿Cómo contribuir con la integración de los procesos de la gestión de servicios en el Centro de Soporte UCI?

Tomando como **objeto de estudio** la integración de los procesos de la gestión de servicios.

Para dar solución al problema planteado se establece como **objetivo general**: Desarrollar las interfaces para el módulo de administración que contribuya con la integración de los procesos de la gestión de servicios en el Centro de Soporte UCI.

Definiendo como **campo de acción** la integración de los procesos de la gestión de servicios en el Centro de Soporte UCI.

Para cumplir con el objetivo general propuesto y darle solución a la situación problemática planteada, se proponen las siguientes **tareas de investigación**:

- 1-Realización del diseño teórico de la investigación.
- 2-Análisis del estado del arte sobre los procesos de gestión de servicio.
- 3-Realización del estudio de las herramientas y tecnología a utilizar en el proceso de desarrollo.
- 4-Realización del análisis y diseño de la solución.
- 5-Realización del modelo de implementación.
- 6-Implementación de los componentes del modelo.
- 7-Diseño y ejecución de los casos de prueba de las funcionalidades implementadas.

Se plantea para la investigación como **Idea a Defender**: El desarrollo de las Interfaces para el Módulo de Administración contribuirá con la integración de los procesos de la gestión de servicios en el Centro de Soporte UCI.

Para darle cumplimiento al objetivo general y respuesta al problema a resolver planteado fueron aplicados los siguientes métodos científicos de investigación:

Teóricos:

Análisis histórico-lógico: Este método se utilizó para entender los antecedentes y tendencias actuales del tema de investigación, así como otras temáticas estrechamente relacionadas con la misma a lo largo de la historia de la Informática.

Analítico–sintético: Este método se utilizó para analizar y comprender la teoría y documentación relacionada con el tema de investigación, permitiendo así, extraer los elementos más relacionados e importantes con el objeto de estudio.

Empíricos:

Observación: Este método es de vital importancia ya que ha permitido percibir a partir de la situación real que se está investigando cómo se desarrolla el proceso que constituye el objeto de estudio. Se utilizó para realizar el diagnóstico acerca de la realización del sistema propuesto.

Entrevista: Para el desarrollo de este método se han entrevistado a especialistas del Centro de Soporte de la Universidad de Ciencias Informáticas, quienes han aportado elementos significativos a la investigación.

El documento está constituido por tres capítulos estructurados de la siguiente forma:

Capítulo 1: Fundamentación Teórica de la investigación: Se definen elementos teóricos que sustentan la investigación, se hace referencia a los conceptos fundamentales para permitir entender cualquier concepto asociado a la comprensión del trabajo, se realiza un estudio del estado del arte de soluciones existentes que puedan dar solución de alguna forma al problema científico planteado y se definen las herramientas y tecnologías a utilizar.

Capítulo2: Propuesta de Solución: Se especifican los requisitos funcionales y no funcionales según necesidades del cliente y a partir de esto se diseña y se implementa el sistema para darle solución a la situación problemática planteada.

Capítulo3: Pruebas del Sistema: Se le realizan pruebas al sistema propuesto y luego se procede a validarlo para verificar que cumple con el objetivo planteado, concluyendo con el resultado de dicha validación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LAS INTERFACES PARA EL MÓDULO DE ADMINISTRACIÓN DEL SISTEMA DE GESTIÓN DE SERVICIOS DE SOPORTE

1.1 Introducción

En este capítulo, se analizan algunos de los conceptos que sustentan los principales elementos de la investigación referentes a proceso, integración y administración. Además se realiza un análisis de las características más significativas y procesos fundamentales que postula ITIL; así como también una breve comparación sobre los diferentes estándares existentes en el mundo, utilizados para la gestión de servicios TI respecto a ITIL, profundizando así en la necesidad de la solución propuesta. Además del estudio sobre el estado del arte de los diferentes Service Desk existentes a nivel nacional e internacional, a través del cual se gestionan los servicios de soporte. Adicionalmente en este capítulo, se exponen las principales herramientas, lenguajes y tecnologías que se utilizarán en el desarrollo de las Interfaces para el Módulo de Administración del Sistema de Gestión de Servicios de Soporte del Centro de Soporte UCI.

1.2 Conceptos Fundamentales

Para dar un mejor entendimiento sobre la problemática a resolver, a continuación se muestran un conjunto de conceptos que constituyen la base teórica para el pleno desarrollo del tema tratado en este trabajo.

Proceso, Integración y Administración

Entender estos términos se ha vuelto engorroso por las diversas definiciones existentes hoy en día, las cuales, varían según el autor. En el presente trabajo se proporciona una definición general de cada uno de ellos que están basadas en las propuestas de algunos prestigiosos autores, con la finalidad de brindar, una idea general del significado de estos términos en nuestros días.

Proceso

Un proceso es una secuencia de tareas o el conjunto de actividades y decisiones para producir un producto final (Dybå Tore, Moe Nils Brede, 2004).

Un proceso es cualquier actividad o grupo de actividades que emplee un insumo, le agregue valor a este y suministre un producto a un cliente externo o interno (Harrington, 1994).

Con lo plasmado anteriormente queda explícito que un proceso no es más que un conjunto de actividades estrechamente relacionadas. Estas actividades requieren el uso de recursos tanto material como personal y su fin es proporcionar resultados para obtener un producto final y satisfacer con este a los clientes.

Integración

La integración consiste en integrar o hacer un todo con las partes; convirtiendo las unidades antes separadas en componentes de un sistema coherente (Deutsch, 1988).

La integración definida verbalmente como lo que constituye partes en un todo o lo que crea interdependencia (Nye, 1968).

Por lo que se puede concluir que el término integración se puede entender como la formación de un todo, o sea completar un todo con las partes que faltaban o hacer que algo pase a formar parte de ese todo.

Administración

La administración es el proceso de planear, organizar, dirigir y controlar el uso de los recursos para lograr los objetivos organizacionales (Idalberto, 2004).

La administración es la coordinación de las actividades de trabajo de modo que se realicen de manera eficiente y eficaz con otras personas y a través de ellas (Stephen Robbins, Mary Coulter, 2005).

Teniendo en cuenta las anteriores propuestas, se podría decir que para la presente investigación administración es:

El proceso de realizar un conjunto de actividades esenciales de forma secuencial y coherente, las cuáles son planificar, organizar, dirigir y controlar. Las mismas están enfocadas a la utilización viable de los distintos tipos de recursos que dispone la organización, ya sea humano, financiero, tecnológico o informativo, así como también al conjunto de tareas o actividades, que al igual que los recursos, son indispensables para lograr los objetivos o metas establecidos o trazados por la organización de una forma eficaz y eficiente.

1.3 Service Desk

Las grandes empresas del mundo dependen cada vez más de tecnologías de alta fiabilidad, para garantizar que los servicios de soporte técnico brindados sean eficientes y tengan la calidad requerida. De forma tal que ningún problema paralice su funcionamiento y establecimiento de una relación sólida y comprometida con el cliente.

Por las razones expuestas anteriormente esta herramienta ha ido cobrando gran importancia en estas entidades, por lo que no podemos avanzar con el desarrollo de este trabajo, sin antes abordar una cuestión tan relevante como la atención y resolución de las necesidades de los usuarios de TI a través del Service Desk.

Antes de mencionar algunos aspectos importantes sobre él mismo, como funciones, tipos y beneficios, es necesario conocer su papel dentro de las organizaciones.

¿Cuál es el papel del Service Desk?

El Service Desk es una función organizativa, no un proceso, dentro del departamento de TI. Es un software que permite a los usuarios responder las demandas de infraestructuras complejas (Pía Ramírez Bravo, Felipe Donoso Jaurés, 2006), pues en él se centraliza la recepción tanto de incidencias como de peticiones de servicio y reclamaciones. *"Además soluciona de una forma eficiente, la gestión de recursos y es el único encargado en garantizar que se encontrará a la persona correcta para ayudar a resolver el problema que se presente"* (Pía Ramírez Bravo, Felipe Donoso Jaurés, 2006). Para avalar que se resuelvan dichas necesidades, tiene que realizar una actividad estrechamente relacionada con la organización, para así mantener informado al usuario del estado de sus solicitudes en todo momento.

Dado que se convierte en el único punto de contacto entre la organización de servicios de TI y los usuarios, está dotado de gran responsabilidad al ser la imagen de todo el departamento y, por tanto, asume un papel protagonista en la satisfacción de los usuarios con los servicios de TI.

Funciones y Actividades

Un Service Desk, tiene otras responsabilidades además de las ya mencionadas y son las relacionadas con aportar más valor a la gestión de servicios. Este software ejerce varias funciones, pero solo se mencionará la principal:

- Gestionar la relación con los clientes y usuarios manteniéndoles puntualmente informado de todos aquellos procesos de su interés (Osiatis, 2010).

Las actividades del Service Desk pueden abarcar de una manera u otra casi todos los aspectos de la Gestión de Servicios TI, a continuación se mencionan algunas de las actividades que debería ofrecer para merecer ese nombre (Osiatis, 2010):

Gestión de Incidentes

Independientemente de que la completa gestión de las incidencias requiera la colaboración de otros departamentos y personal, el Service Desk debe ofrecer una primera línea de soporte para la solución de todas las interrupciones de servicio y/o peticiones de servicio que puedan cursar los clientes y usuarios.

Entre sus tareas específicas se incluyen:

- Registro y monitorización de cada incidente.
- Comprobación de que el servicio de soporte requerido se incluye en el SLA asociado.
- Seguimiento del proceso de escalado.
- Identificación de problemas.
- Cierre del incidente y confirmación con el cliente.

Centro de información

El Service Desk debe ser la principal fuente de información de los clientes y usuarios, informando sobre:

- Nuevos servicios.
- El lanzamiento de nuevas versiones para la corrección de errores.
- El cumplimiento de los SLAs.

Este contacto directo con los clientes debe servir también para identificar nuevas oportunidades de negocio, evaluar las necesidades de los clientes y su grado de satisfacción con el servicio prestado.

Esta herramienta se encuentra en una situación inmejorable para ofrecer también información privilegiada a todos los procesos de gestión de los servicios TI. Es para ello imprescindible que se lleve un adecuado registro de toda la interacción con los usuarios y clientes.

Relaciones con los proveedores

El Service Desk es también responsable de la relación con los proveedores de servicios de mantenimiento externos.

Es imprescindible, para ofrecer un servicio de calidad, una estrecha relación entre los responsables externos del mantenimiento y la Gestión de Incidentes que debe ser canalizada a través del mismo.

Tipos

Esta tecnología se ha ido desarrollando, ganando en complejidad y funcionalidad, y en dependencia de la profundidad de los servicios prestados a los usuarios, el punto de gestión varía, dentro de los más comunes se encuentran (Riverón, 2007):

- **Call Center** (Centro de Llamadas): Es el más sencillo de todos y gestiona una gran cantidad de llamadas telefónicas en función de brindar soporte, redireccionando a los usuarios a otras instancias de soporte.
- **Help Desk** (Mesa de Ayuda): Maneja y coordina los incidentes de la manera más rápida, ayudando a incrementar la productividad y aumentando la satisfacción de los usuarios internos y externos.
- **Service Desk** (Mesa de Servicio): Incluye además de la funcionalidad del Help Desk (gestión de incidentes) la gestión de problemas, consultas; así como otros servicios que facilitan el negocio.

En la figura 1 Tipos de Service Desk según su estructura física se pueden evidenciar los 4 tipos fundamentales que existen de esta otra clasificación (Customer Care Associates, 2011).

- **Centralizado:** Implica la atención desde un solo punto a usuarios ubicados en diferentes localidades. Adecuado para algunas organizaciones y tipos de soporte, origina costos más bajos y evita la duplicidad de instalaciones, habilidades y procedimientos.
- **Distribuido o local:** Proporciona atención en la misma localidad/ubicación de los usuarios. Adecuado para organizaciones con sucursales en varios lugares que requieren soporte personal inmediato.
- **Virtual:** Ubicado en cualquier lugar del mundo, con un proveedor que brinde soporte desde las oficinas en alguna otra parte. Soporte para aplicaciones de software, a través de Internet, telecomunicaciones avanzadas y tecnologías de red.
- **Mixto:** Combinación de dos o más de los tipos de Service Desk antes mencionados, para una flexibilidad máxima.

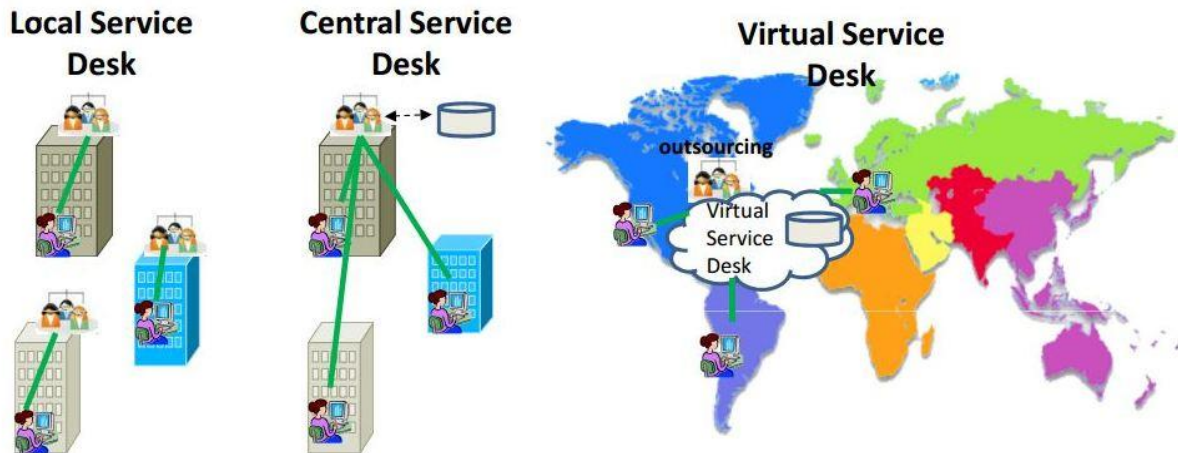


Figura 1. Tipos de Service Desk según su estructura física.

Son muchos los beneficios que se adquieren utilizando esta tecnología, pero específicamente el Service Desk, arroja óptimos resultados sobre la mejora del servicio y la satisfacción del usuario.

Con el análisis de esta importante tecnología, se llegó a la conclusión de que son muchas las razones por las que un Service Desk va a ser beneficioso, pero la cuestión no es: ¿Por qué utilizar un Service Desk? sino ¿Cómo todavía no se ha pensado en utilizarlo?

1.4 Estándares para la gestión de servicios TI

Existen disímiles estándares o marcos de referencias de gran aceptación mundial y la lista seguirá creciendo. Se deben analizar y seleccionar aquellos que mejor se adapten a los objetivos y necesidades de negocios de cada organización con el fin de ayudar a la empresa a cumplir los mismos, única razón para usarlos y realizar una integración entre ellos. Dentro de los más importantes se destacan: COBIT, CMMI-SVC, ISO/IEC 20000 e ITIL.

1.4.1 COBIT

COBIT habilita el desarrollo de una política clara y de buenas prácticas de control de TI a través de organizaciones, a nivel mundial. Fue creado por la Asociación para la Auditoría y Control de Sistemas de Información (ISACA) y el Instituto de Administración de las Tecnologías de la Información (ITGI) en 1992 (Instituto de Gobierno de TI, 2008). Su misión es proporcionar objetivos de control, dentro del marco

referencial definido, y obtener la aprobación y el apoyo de las entidades comerciales, gubernamentales y profesionales en todo el mundo.

Por lo tanto, está orientado a ser la herramienta de gobierno de TI que ayude al entendimiento y a la administración de riesgos asociados con tecnología de información y con tecnologías relacionadas.

1.4.2 ISO/IEC 20000

La ISO/IEC 20000 proporciona una norma internacional para todas las empresas que ofrezcan servicios de TI tanto a clientes internos como externos (Layme Velázquez, Rubén Darío, 2011). La especificación y los requisitos que esta plantea representan un consenso para la industria en estandarización de calidad para la gestión de los servicios TI. Además promueve la adopción de un enfoque de procesos que permiten satisfacer las necesidades del negocio y del cliente. Este estándar está orientado a procesos proporcionando calidad y un mejor servicio.

Es válido aclarar que la certificación ISO/IEC 20000 sólo se otorga a organizaciones que realizan operaciones de gestión de servicios TI, y que la norma sólo certifica el buen funcionamiento de esas operaciones, por lo tanto no entran en su ámbito de competencia la certificación de productos, ni servicios de consultoría relativos a la aplicación de buenas prácticas.

1.4.3 CMMI-SVC

El CMMI-SVC es una nueva constelación dentro de la suite de CMMI que pretende cubrir las necesidades de la industria de servicios. Reconociendo, que estos son un catalizador en el crecimiento económico mundial y la necesidad de establecer un marco de referencia que proporcione una guía para desarrollar y mejorar las prácticas de provisión de servicios de las organizaciones, como un medio para mejorar la satisfacción del cliente, el desempeño y la rentabilidad de las organizaciones que proveen servicios (Avantare, 2014).

Este estándar, se centra en la provisión de servicios de calidad a los clientes y usuarios finales. Conteniendo prácticas que cubren los procesos de administración de proyectos, administración de procesos, establecimiento de servicios, entrega de servicios; así como otros procesos de soporte. El mismo, está orientado al proceso, pero se enfoca en el desarrollo.

1.4.4 ITIL

ITIL es un estándar enfocado en la gestión de servicios informáticos que es usado para lograr eficacia y eficiencia en los servicios TI. Su principal objetivo es mejorar la calidad de estos, evitar los problemas asociados a los mismos y en caso de que estos problemas ocurran, ofrecer un marco de actuación para que estos sean solucionados con el menor impacto y a la mayor brevedad posible (Osiatis, 2010). Fue desarrollado al reconocer que las organizaciones dependen cada vez más de la Informática para alcanzar sus objetivos corporativos. Además ha demostrado ser útil para las organizaciones en todos los sectores a través de su adopción por innumerables compañías como base para consulta, educación y soporte de herramientas de software.

Comparación de ITIL con COBIT

COBIT está enfocado en el control (métricas e indicadores de rendimiento) y es usado para establecer mediciones y aplicar correctivos, mientras ITIL es un estándar enfocado en la gestión de servicios informáticos que es usado para lograr eficacia y eficiencia en los servicios TI.

COBIT puede que tenga mayor alcance que ITIL ya que abarca todo el espectro del Gobierno de TI, sin embargo ITIL está centrado solo en gestión de servicios.

Ambas guías son también complementarias y se pueden usar juntas: COBIT para verificar la conformidad en cuanto a disponibilidad, rendimiento, eficiencia y riesgos asociados de dichos servicios con los objetivos y estrategias de la compañía, usando para ello métricas claves y cuadros de mando que reporten dicha información e ITIL para lograr efectividad y eficiencia en los servicios TI.

Comparación de ITIL con ISO/IEC 20000

Como se puede apreciar, no todos los estándares son aplicables al área de gestión de servicios de TI. De ellos solamente la ISO 20000 e ITIL se enfocan a esta área. La diferencia más significativa entre ambos, es que en el caso de la ISO 20000, se certifica como tal a las organizaciones que buscan mejorar sus servicios TI, mediante la aplicación efectiva de los procesos para monitorizar y mejorar la calidad de los servicios que brindan mientras que con ITIL sólo se certifican a las personas. Además este estándar se enfoca solamente en los procesos para satisfacer las necesidades del negocio y del cliente mientras que ITL se enfoca en la gestión de los servicios de tecnología de la información.

Comparación de ITIL CMMI-SVC

CMMI ha demostrado ser una metodología de gran eficacia, que ha permitido mejoras de gran magnitud en varios procesos de desarrollo de productos software. Sin embargo su falta de adecuación del enfoque al servicio que está experimentando el sector de las TI y el alto esfuerzo de implantación que exige, está siendo un problema para cualquier organización. En cambio ITIL garantiza la explotación del producto software y está enfocado sólo en gestión de servicios, con el propósito de lograr eficacia y eficiencia en los mismos.

Ambos modelos, son complementarios y se pueden utilizar juntos en proyectos de análisis y definición de procesos que permitan encajar ambas filosofías de trabajo.

Análisis crítico de los estándares analizados.

Según el análisis y estudio realizado sobre estas guías de mejores prácticas, se pudo concluir que para el desarrollo de la solución propuesta de este trabajo se aplicará el marco de referencia ITIL. Este marco se adapta perfectamente a las necesidades, objetivos y metas del centro de soporte de la Universidad de las Ciencias Informáticas. Además los factores principales que influyeron en su selección fueron: su enfoque, el cual va dirigido únicamente a la gestión de servicios de TI, permitiendo apoyar la prestación de los mismos a través de las pautas que establece como guía que implementa buenas prácticas. Además brinda la posibilidad de poderse integrar con otros estándares para así fortalecer cada uno de sus procesos, de ser necesario.

1.5 ITIL

“ITIL es mundialmente reconocido por sus siglas en inglés, pues constituye la aproximación más ampliamente aceptada en el mundo para la gestión del servicio TI” (OSIATIS, 2010). Por lo que ha llegado a ser considerado un estándar de facto para la gestión de servicios en el mundo. Fue creado a finales de los 80 y desarrollado a petición del Gobierno del Reino Unido.

Actualmente es una marca registrada de la Oficina de Comercio del Gobierno Británico OGC (*Office of Government Commerce*), y es de libre utilización para los sectores tanto públicos como privados.

“Este marco de referencia se basa en la calidad del servicio y el desarrollo eficaz y eficiente de los procesos que cubren las actividades más importantes de las organizaciones. Proponiendo una terminología estándar e independiente de la industria y la tecnología, para definir “qué hacer” y “qué no

hacer” al aplicar en una organización la administración de servicios de las TI desde el punto de vista del negocio. Garantizando de esta forma los niveles de servicio establecidos entre la organización y sus clientes” (Pía Ramírez Bravo, Felipe Donoso Jaurés, 2006). Para ello, apoya, más no dicta los procesos de negocios en una organización, por lo que sus mejores prácticas adquieren distintas formas y matices, adaptándose a las necesidades individuales de cada entidad.

Actualmente esta guía de mejores prácticas ha sido reestructurada, dándole paso a la publicación de una nueva edición de la librería ITIL, llamada ITIL 2011, la cual no es una versión de ITIL, como lo fue la versión 3, sino que es una actualización de esta. Por lo que sigue manteniendo el conjunto de publicaciones definidas en ITIL v3, las cuales se corresponden con las 5 fases que postula dicho estándar (Osatis, 2010):

- Estrategia del Servicio: Propone tratar la gestión de servicios no sólo como una capacidad sino como un activo estratégico.
- Diseño del Servicio: Cubre los principios y métodos necesarios para transformar los objetivos estratégicos en portafolios de servicios y activos.
- Transición del Servicio: Cubre el proceso de transición para la implementación de nuevos servicios o su mejora.
- Operación del Servicio: Cubre las mejores prácticas para la gestión del día a día en la operación del servicio.
- Mejora Continua del Servicio: Proporciona una guía para la creación y mantenimiento del valor ofrecido a los clientes a traves de un diseño, transición y operación del servicio optimizado.

“ITIL se centra en brindar servicios de alta calidad para lograr la máxima satisfacción del cliente a un costo manejable. Para ello, parte de un enfoque estratégico basado en el triángulo procesos-personas-tecnología. En otras palabras: determina la forma de ejecutar procesos estándar ayudados de la tecnología para lograr la satisfacción de las personas, usuarios de los servicios de TI” (Coello, 2008). La figura 2 Enfoque estratégico de ITIL muestra dicho enfoque.



Figura 2. Enfoque Estratégico de ITIL.

“A través de este enfoque, ITIL ofrece un método probado para gestionar procesos, roles y actividades, así como sus interrelaciones” (OSIATIS, 2010). Puede emplearse en organizaciones que ya tengan sus propios métodos y actividades de gestión de servicios independientemente de su tamaño.

Este marco de referencia estructura la gestión de los servicios TI sobre el concepto de Ciclo de Vida de los Servicios. “*Este enfoque tiene como objetivo ofrecer una visión global de la vida de un servicio desde su diseño hasta su eventual abandono sin por ello ignorar los detalles de todos los procesos y funciones involucrados en la eficiente prestación del mismo*” (OSIATIS, 2010).

El Ciclo de Vida del Servicio consta de cinco fases. La figura 3 muestra los diferentes procesos oficiales que componen dichas fases, contenidos en esta edición:

Estrategia del Servicio	Diseño del Servicio	Transición del Servicio	Operación del Servicio	Mejora Continua
1. Gestión estratégica para los servicios de TI	1. Coordinación del Diseño	1. Planeación y Soporte a la Transición	1. Gestión de Eventos	1. Mejora de los 7 pasos
2. Gestión del Portafolio de Servicios	2. Gestión del Catálogo de Servicios	2. Gestión de Cambios	2. Gestión de Incidentes	
3. Gestión Financiera para Servicios de TI	3. Gestión de Niveles de Servicio	3. Gestión de Activos de Servicio Configuraciones	3. Cumplimiento de Solicitudes	
4. Gestión de la Demanda	4. Gestión de Disponibilidad	4. Gestión Liberaciones e Implementaciones	4. Gestión de Problemas	
5. Gestión de Relaciones con el Negocio	5. Gestión de Capacidad	5. Validación y Pruebas del Servicio	5. Gestión de Accesos	
	6. Gestión de Continuidad de Servicios de TI	6. Evaluación del Cambio		
	7. Gestión de seguridad de la Información	7. Gestión del Conocimiento		
	8. Gestión de Proveedores			

Figura 3. Procesos oficiales de ITIL en correspondencia con sus fases. (Corona, 2011)

Los procesos de Gestión de Servicios de ITIL, han sido concentrados en una tercera versión y posteriormente en la reciente edición 2011 mencionada anteriormente. Esta edición goza de mayor uso en nuestros días, pues organiza su librería en dos conjuntos significativos, los cuales son: El Soporte de Servicio y La Prestación de Servicio. *"Los procesos involucrados en estas dos áreas, proporcionan control total sobre las necesidades, requerimientos, desafíos en el negocio, clientes y los usuarios finales"* (Sun Microsystems, 2003).

1.5.1 Metodología que propone ITIL para brindar servicios de soporte

Los procesos de Gestión de Servicios son el corazón de ITIL y pueden subdividirse en 2 áreas bien diferenciadas, mencionadas anteriormente, las cuáles son: El Soporte de Servicio y La Prestación de Servicio.

El Soporte a los Servicios generalmente se concentra en las operaciones cotidianas, así como en dar soporte a los servicios de TI, preocupándose de todos los aspectos que garanticen la continuidad, disponibilidad y calidad del servicio prestado al usuario. En cambio, La Prestación de Servicios se ocupa de la planificación a largo plazo y del perfeccionamiento de la provisión de estos servicios (Sun

Microsystems, 2003). La tabla 1 mostrada a continuación, destaca los procesos que manejan estas dos áreas. Sin embargo, en el desarrollo del análisis de este subepígrafe, se hará énfasis en los procesos del área de El Soporte de Servicio.

Soporte de Servicio	Prestación de Servicio
Service Desk	
Gestión de Incidentes	Gestión de Nivel de Servicio
Gestión de Problemas	Gestión Financiera para Servicios de TI
Gestión de Cambios	Gestión de la Capacidad
Gestión de Versiones	Gestión de la Continuidad
Gestión de Configuración	Gestión de la Disponibilidad

Tabla 1. Procesos de las áreas Soporte de Servicio y Prestación de Servicio.

Dentro de los procesos de Soporte de Servicios, ITIL recomienda la implementación de una función de Service Desk, en el ámbito operativo, a fin de proporcionar un único punto de contacto para brindar información, asistencia y una rápida restauración del servicio a clientes y usuarios. Al ser un punto de contacto inicial, reduce la cantidad de trabajo a otras áreas de TI, interceptando preguntas irrelevantes y aquellas que son fáciles de contestar. De manera general es el encargado de centralizar el manejo de incidentes, solicitud de cambios, solución de problemas, integración de nuevas versiones así como realizar el apropiado direccionamiento y escalamiento en caso de no contar con una solución inmediata a las dificultades que se presenten.

Este centro de servicios, en su concepción más moderna, debe representar el centro neurálgico de todos los procesos de soporte al servicio (OSIATIS, 2010):

- Registrando y monitorizando incidentes.
- Aplicando soluciones temporales a errores conocidos en colaboración con la Gestión de Problemas.

- Colaborando con la Gestión de Configuraciones para asegurar la actualización de las bases de datos correspondientes.
- Gestionando cambios solicitados por los clientes mediante peticiones de servicio en colaboración con la Gestión de Cambios y Versiones.

Pero también debe jugar un papel importante dando soporte al negocio, identificando nuevas oportunidades en sus contactos con usuarios y clientes.

Los procesos de soporte al servicio presentan un grupo de objetivos, como es el caso de (OSIATIS, 2010):

- Gestión de Incidentes: Este proceso tiene como objetivo resolver cualquier incidente que cause una interrupción en el servicio de la manera más rápida y eficaz posible, restableciendo la operación al servicio normal sin mayores retardos y causando el menor impacto posible en las operaciones del negocio.
- Gestión de Problemas: Elimina los incidentes y los errores repetitivos, investigando las causas subyacentes de toda alteración, real o potencial y determinando posibles soluciones.
- Gestión de Cambios: Asegura que se utilizan métodos y procedimientos estandarizados para la evaluación y planificación del proceso de cambio de manera que, si este se lleva cabo, se haga de la forma más eficiente, siguiendo los procedimientos establecidos y asegurando en todo momento la calidad y continuidad del servicio.
- Gestión de Versiones: Planea y supervisa la introducción exitosa de software y hardware además de diseñar e implementar procedimientos eficientes para la distribución e instalación de cambios a los sistemas de TI.
- Gestión de Configuración: Controla todos los elementos de la Infraestructura de TI con el nivel de detalle o profundidad que mejor convenga al sistema, además de encargarse de verificar los registros de configuración, corrigiendo cualquier omisión si fuese necesario.

Otros conceptos importantes que se integran a estos procesos son (OSIATIS, 2010):

- Clientes: Los clientes van a ser los encargados de contratar los servicios TI. Además son las personas a las que se les rinde cuenta respecto a los Niveles de Acuerdo de Servicios.

Capítulo 1. Fundamentación Teórica

- Usuarios: Los usuarios son aquellos que utilizan los servicios TI para llevar a cabo sus actividades.
- Organización: La propia organización TI debe considerarse como otro cliente/usuario más de los servicios TI.
- La Base de Conocimiento (Knowledge Base, KB, por sus siglas en inglés): Esta base de conocimiento debe recoger toda la información necesaria para:
 - Ofrecer una primera línea de soporte ágil y eficaz sin necesidad de recurrir a escalados.
 - Realizar una tarea comercial y de soporte al negocio.

La figura 4 resume concisamente los principales aspectos de la metodología de soporte al servicio según los estándares ITIL:

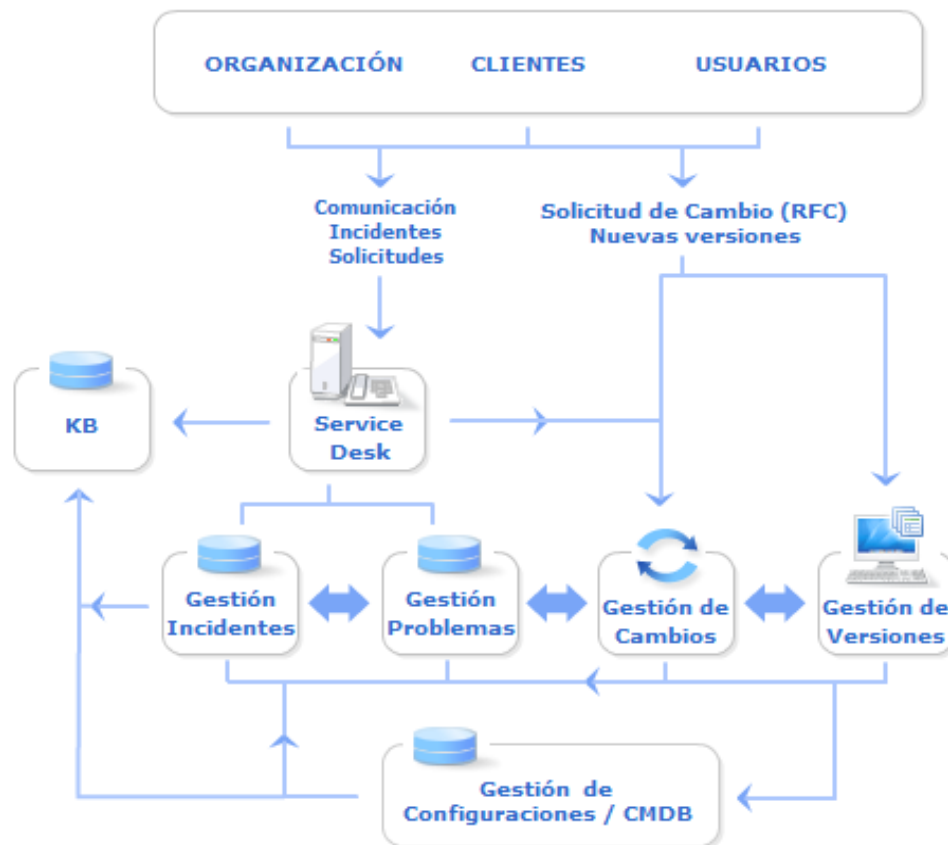


Figura 4. Principales aspectos de la metodología de soporte al servicio. (OSIATIS, 2010)

1.6 Análisis de soluciones existentes

Antes de comenzar a desarrollar la solución propuesta, se realizó un análisis de varias soluciones existentes a nivel nacional e internacional con el mismo fin, para obtener las principales características y ventajas que podrían aportar las mismas a la solución propuesta. Además, el mencionado análisis permitió definir el porqué de la solución propuesta. A continuación se detallan algunas de las características fundamentales de las soluciones analizadas en el ámbito internacional.

1.6.1 NetSupport ServiceDesk

Es un potente gestor de incidencias basado en web, el cual fue desarrollado por la empresa NetSupport. A través de la automatización de las tareas repetitivas y la gestión y control de las incidencias ha ayudado a miles de empresas a dar soporte a sus usuarios con menos recursos y más eficiencia. La mayoría de estas incidencias que se presentan, son repeticiones o variaciones sobre problemas frecuentes (TotemGuard Digital Security, 2012).

Las características más significativas de este sistema, que le permite ser más utilizado son (TotemGuard Digital Security, 2012):

- Gestionar incidencias según las buenas prácticas de ITIL.
- Permite crear una base de conocimiento y dejar de solucionar las mismas incidencias una y otra vez.
- Automatizar el registro, asignación y respuesta de incidencias.
- Generar informes con estadísticas de la eficacia del departamento.

1.6.2 CA Service Desk Manager

Es una solución versátil y completa de soporte de servicio de TI, el cual fue desarrollado por la empresa CA Technologies. Esta herramienta web permite evitar interrupciones en el servicio, gestionar mejor los riesgos vinculados al cambio y proporcionar una visión completa de los servicios de TI. Fue diseñada para ayudarle a proporcionar al usuario final, una asistencia técnica de gran calidad a través de una estrecha integración con los procesos de gestión de incidencias, problemas, conocimiento, activos de servicio y configuración.

Las características más significativas son (CA technologies, 2012):

- Diseñado para implementar mejores prácticas de servicio de TI (ITIL).

- Análisis avanzado de la causa raíz de los problemas.
- Herramientas de creación de reportes.
- Integra base de datos de gestión de configuración (CMDB).
- Compatible con sistemas Windows, Unix o Linux.

1.6.3 Servicio de Referencia Virtual y Call Center

En Cuba, la tecnología Service Desk no ha tenido mucho auge, lo más cercano que se ha desarrollado es la gestión de información y conocimiento a través del Servicio de Referencia Virtual y algunos Call Center implementados por ETECSA.

"Un ejemplo es el proyecto Pregúntale al Bibliotecario implementado en el 2003, en la biblioteca de la Universidad central de las Villas, Marta Abreu. El mismo se basa en el servicio tradicional que se presta en una biblioteca a través de un módulo de referencia en un ambiente virtual, para ofrecer un servicio ágil y de calidad en cualquier horario. El ambiente virtual reúne todas las facilidades que un usuario puede recibir mediante un buró de referencia.

En cambio ETECSA, para su funcionamiento implementa la tecnología Call Center o Centros de Atención al Cliente, ideados con el objetivo de ofrecer información relacionada con la telefonía a nivel nacional e internacional. Para lograrlo, cuenta con un equipo de operadoras calificadas, variedad de servicios y la más avanzada tecnología. Como todo Service Desk, los Call Center de ETECSA son evaluados por el grado de satisfacción de usuarios y clientes con el servicio prestado" (Riverón, 2007).

En la Universidad de las Ciencias Informáticas existe un Service Desk, basado en la tecnología web, conocido como NovaDesk. Surge como una alternativa para facilitar la calidad del trabajo de cada usuario y enfrentar de manera más, rápida y eficiente el nuevo sistema operativo NOVA.

1.6.4 ManageEngine Service Desk Plus

Otro ejemplo del estudio es el Service Desk usado en el Centro de Soporte de la UCI denominado ManageEngine Service Desk Plus, a través del cual se brinda soporte a todos los servicios y aplicaciones desarrolladas por los centros productivos. "ServiceDesk Plus es un completo sistema Service Desk, desarrollado por la empresa ManageEngine. Está basado en las mejores prácticas de ITIL y se caracteriza por ser escalable, modular y de fácil adquisición. Además es uno de los sistemas Service Desk más populares del mercado y cuenta con miles de usuarios en todo el mundo." (Ireo, 2002).

Entre sus amplias prestaciones, se destacan (Ireo, 2002):

- Gestión de incidencias y solicitudes.
- Gestión y seguimiento de tareas.
- Gestión de configuraciones.
- Gestión de problemas.
- Gestión de cambios.
- Catálogo de servicios.
- CMDB.
- Gestión y seguimiento de pedidos.
- Inventario automático HW y SW y gestión de activos.
- Gestión de contratos, mantenimientos y garantías.
- Portal de Autoservicio para usuarios finales.
- Base de conocimientos.
- Sistema de SLAs, reglas de negocio y workflow personalizable.
- Interfaz muy personalizable.
- Avisos y alertas, notificaciones automáticas, informes personalizables.

Análisis crítico de las soluciones analizadas

Los sistemas internacionales estudiados poseen varias características y funciones que son similares a las del Sistema de Gestión de Servicios, que actualmente se desarrolla en el centro de Soporte de la UCI. Sin embargo, la adquisición de estos se vuelve engorrosa, debido a que son costosos y a que el país se encuentra en la migración hacia el software libre. Luego de analizar las soluciones existentes en nuestro país, se denotó que la solución que brindan no es suficiente. Estas no se adaptan a las necesidades reales del cliente, pues sus funcionalidades son escasas. Tras el análisis realizado, se llega a la conclusión, que es de suma importancia el desarrollo de una herramienta nacional. Tomando como guía las características semejantes que poseen los sistemas estudiados. Esta herramienta, debe ser capaz de permitir la futura gestión e integración tanto de los procesos involucrados en el Sistema de Gestión de Servicios para el Centro de Soporte UCI como de los módulos de Gestión de Incidencias, Problemas, Cambios, Disponibilidad, Catálogo, Niveles de Servicios y la Base de Datos de Gestión de Configuraciones (*CMDB, por sus siglas en inglés*), pues el sistema existente en dicho centro, a pesar de

proporcionar una amplia gama de funcionalidades y beneficios, el desempeño del equipo de trabajo del centro se ve afectado, debido a varios factores que afectan la calidad de los servicios prestados.

1.7 Metodología de Desarrollo

Una metodología es un entorno que se usa para estructurar, planificar y controlar el proceso de desarrollo de un software (Romero, 2009).

Estas se clasifican en:

- Metodologías Pesadas: "Orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas y notaciones a utilizar" (Isaías Carrillo Pérez, Rodrigo Pérez González, 2008).
- Metodologías ligeras/ágiles: "Orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando" (Isaías Carrillo Pérez, Rodrigo Pérez González, 2008).

"El ciclo de vida ideal de XP consiste en cuatro fases fundamentales: Planificación, Diseño, Desarrollo y Prueba. Se trata de realizar ciclos de desarrollo cortos (llamados iteraciones), con entregables funcionales al finalizar cada ciclo. En cada iteración se realiza un ciclo completo de análisis, diseño, desarrollo y pruebas" (DREW, 2012).

En el desarrollo del sistema se utilizará la metodología Programación Extrema (Extreme Programming, XP, por sus siglas en inglés), que se encuentra dentro de las metodologías ágiles. Esta se basa en la simplicidad, la comunicación y el reciclado del código, aspectos fundamentales a tener en cuenta para el desarrollo de la solución propuesta. XP es muy acertada ya que es ligera, mantiene una fuerte comunicación con el cliente y genera menor volumen de documentación, lo que en el caso de proyectos que tengan un corto plazo de entrega constituye una ventaja.

1.8 Herramientas y tecnologías

A continuación se describen las principales herramientas, tecnologías y lenguajes, que fueron utilizadas por políticas del Centro de Soporte UCI.

1.8.1 IntelliJIDEA 12.2.3

"IntelliJ IDEA es un IDE ideal para el desarrollo ágil y es considerado el más inteligente de Java. Esta herramienta se centra en el aumento de su productividad al proporcionar la asistencia de código inteligente para todos los lenguajes como Java, Groovy, PHP, SQL incluyendo PostgreSQL, MySQL y muchos otros dialectos compatibles así como también ofrece soporte avanzado a los marcos y estándares web más importantes como Grails, Struts y otros" (JetBrains, 2014). Además posee y proporciona un conjunto de características y herramientas integradas que hacen que el desarrollo sea más productivo.

1.8.2 Apache Tomcat 2.1.1

La publicación de servicios web u otros servicios remotos requieren ser desplegados en un servidor de aplicaciones web, con este objetivo se utilizará Apache Tomcat durante el desarrollo, por su amplia integración con IntelliJIDEA.

Apache Tomcat es una implementación de código abierto de software de las tecnologías de JavaServlet y JavaServer Pages. Se utiliza para crear desde simples sitios de un servidor hasta redes de grandes empresas y se desarrolla en un entorno abierto y participativo.

1.8.3 Groovy

"Groovy es considerado como una excelente versión de Java. Proporciona un lenguaje altamente expresivo y totalmente orientado a objetos para la creación de aplicaciones Grails. Permitiendo al desarrollador transmitir ideas de forma concisa, natural y de una manera que es fácil de leer y entender" (BRESLAV, 2000). Los desarrolladores con su uso, pueden apreciar su sintaxis familiar y su perfecta integración Java, sabiendo que pueden recurrir a cualquier clase Java desde Groovy y viceversa.

1.8.4 Grails 2.1.1

"Grails es un framework de desarrollo de aplicaciones web para la plataforma Java que se basa en el lenguaje dinámico Groovy. Permitiendo crear aplicaciones en días, en lugar de semanas. En comparación con el típico framework de Java, con este framework se necesita menos código para obtener el mismo resultado. Por lo que menos código significa menos errores y menos líneas de código de mantener. Este funciona bajo un modelo conocido como MVC o "modelo vista controlador", en el que principalmente lo que se hace es crear "Controladores" que son como servicios que manipulan la aplicación web, y todo el código de estos controladores es ejecutado en el servidor web" (Villalobos, 2011).

1.8.5 PostgreSQL 9.2

PostgreSQL es un gestor de bases de datos de código abierto avanzado. Ofrece control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación incluyendo Java y otros.

1.8.6 Bootstrap 2.2.1

Bootstrap es una colección de herramientas de software libre para la creación de sitios y aplicaciones web. Contiene plantillas de diseño basadas en HTML y CSS con tipografías, formularios, botones, gráficos, barras de navegación y demás componentes de interfaz, así como extensiones opcionales de JavaScript. Fue desarrollado por Mark Otto y Jacob Thornton de Twitter, como un framework para fomentar la consistencia a través de herramientas internas. Es de código abierto, además de ser compatible con la mayoría de los navegadores web.

1.8.7 Visual Paradigm 5

Esta herramienta CASE (*Computer Aided Software Engineering*) en español Ingeniería de Software Asistida por Ordenador, es utilizada para modelar el ciclo de vida completo del desarrollo de software: modelación, diseño, construcción, pruebas y despliegue. Soporta UML 2.0 en adelante, entorno de creación de diagramas para BPMN, ingeniería inversa, generación de código, generador de informes, editor de figuras. Permite la integración con otros programas como el IntelliJ IDEA, NetBeans y otros. Se encuentra disponible para los principales sistemas operativos.

1.8.8 JMeter

JMeter, una herramienta de código abierto escrita en Java para realizar pruebas de carga y de rendimiento, principalmente a aplicaciones web.

1.9 Conclusiones Parciales

El estudio de los conceptos más esenciales asociados al tema de investigación garantizó un mejor entendimiento sobre el desarrollo del mismo. Se realizó un análisis del estado del arte tanto a nivel nacional como internacional referente a Service Desk, lo cual permitió llegar a la conclusión de desarrollar las Interfaces de un Módulo de Administración basado en el estándar ITIL para el Sistema de Gestión de Servicios que actualmente se desarrolla en el Centro de Soporte de la UCI. El análisis de la metodología

de desarrollo a utilizar, proporcionó elementos significativos para la implementación del sistema, tales como sus fases y que artefactos son necesarios generar en cada una de ellas para la descripción de la solución propuesta. Para el desarrollo de dicha solución, se utilizarán las herramientas y tecnologías que fueron asignadas por políticas del Centro de Soporte, las cuales garantizarán la futura integración de los módulos que constituirán la plataforma para la Gestión de los Servicios del Centro.

Capítulo 2: Propuesta del Sistema.

2.1 Introducción

En este capítulo se describirá la propuesta de solución a partir del uso de la metodología de Programación Extrema. Se definirán los requerimientos no funcionales, así como los funcionales, basándose en las restricciones y características específicas de la solución. También se tienen en cuenta las fases según la metodología XP y los artefactos generados en cada una de ellas.

2.2 Módulo de Administración para el Sistema de Gestión de Servicios de Soporte de la UCI

Este sistema debe gestionar los procesos involucrados en el Sistema de Gestión de Servicios de Soporte y permitir una futura integración de los módulos de Gestión de Incidencias, Problemas, Cambios, Disponibilidad, Catálogo, Niveles de Servicios y la CMDB, como se muestra en la figura 5 Integración de módulos.

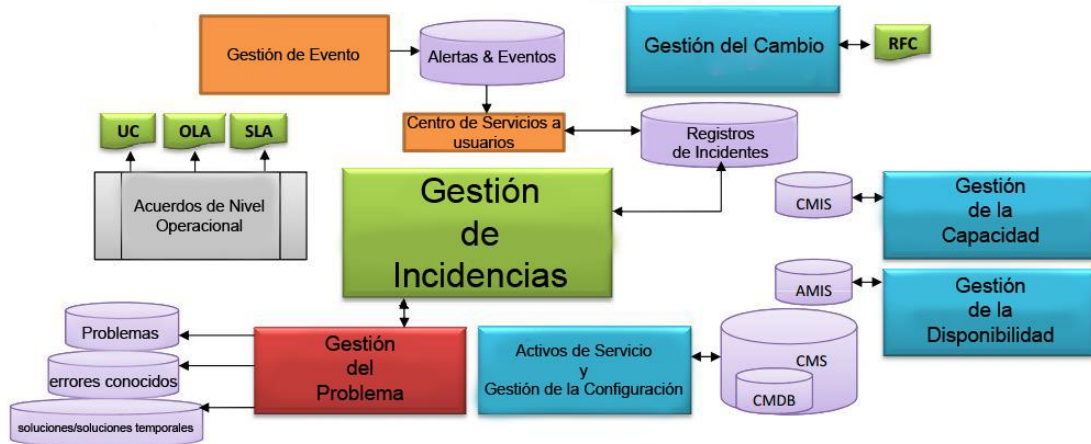


Figura 5. Integración de los módulos.

2.3 Requisitos del sistema

La captura de requerimientos cumple un papel esencial en el proceso de desarrollo de software, se basa en la comunicación entre los desarrolladores, clientes y usuarios, para definir el nuevo sistema que brinde una solución al problema. Su tarea principal consiste en la generación de especificaciones correctas que describan con claridad el comportamiento del sistema, de esta forma, se pretende minimizar los problemas relacionados con el desarrollo del sistema. Estos requerimientos están contenidos en la Lista

de Reserva del Producto, sus siglas en español (LRP). Esta lista se encarga de dar a conocer que es lo que quiere el cliente específicamente. En ella se registrará toda la información sobre el módulo a desarrollar, es una especificación de lo que debe ser implementado.

Requisitos funcionales del sistema

Los requisitos funcionales del sistema son la condición o capacidad que el usuario necesita para resolver un problema o resolver un objetivo. Es la condición o la capacidad que tiene que ser alcanzada o poseída por un sistema o componente de sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente. A continuación se exponen los requisitos funcionales del Módulo de Administración para el Sistema de Gestión de Servicios de Soporte.

Fecha	Versión	Descripción	Autor
9/3/2014	1.0	Confección del documento	Macdemis Liliana Pozo

Código	Descripción	Prioridad	Estimación	Estimado por
RF 1	Autenticar usuario.	Muy alta	4 días	Analista
RF 2	Gestionar usuario.	Muy alta	5 días	Analista
RF 3	Gestionar rol	Muy alta	5 días	Analista
RF 4	Gestionar las tareas de los usuarios en el Sistema de Gestión de Servicios de Soporte.	Muy Alta	5 días	Analista
RF 5	Diseñar interfaz del Sistema de Gestión de Servicios de Soporte.	Alta	4 días	Analista
RF 6	Diseñar interfaz del Módulo de Administración.	Alta	5 días	Analista
RF 7	Diseñar interfaz de la funcionalidad Comentarios.	Alta	3 días	Analista
RF 8	Mostrar ayuda del Sistema de Gestión de Servicios de Soporte.	Media	3 días	Analista
RF 9	Mostrar datos del Sistema de Gestión de Servicios de Soporte.	Media	3 días	Analista

Tabla 2. Lista de reserva del producto. Requisitos funcionales.

Requisitos no funcionales del sistema

Los requerimientos no funcionales (RNF) son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Normalmente, están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. A continuación se exponen los requisitos no funcionales del sistema a desarrollar.

Requisitos No Funcionales		
Usabilidad		
	RNFU 1	El sistema debe brindar un acceso fácil y rápido, para facilitar el uso del mismo por usuarios con pocos conocimientos en el campo de la informática.
Diseño e Implementación		
	RNFDI 1	Se utilizará PostgreSQL en su versión 8.4 o superior como Sistema Gestor de Bases de Datos, dado que el cliente así lo exige.
	RNFDI 2	Se utilizará Groovy como Lenguaje de Programación y Grails como Framework Integrador, dado que el cliente así lo exige.
	RNFDI 3	Se utilizará como arquitectura del sistema Modelo-Vista-Controlador, dado que el cliente así lo exige.
	RNFDI 4	Se utilizará IntelliJIDEA como IDE para el desarrollo del sistema, dado que el cliente así lo exige.
	RNFDI 5	Se utilizará Windows 7 o superior como Sistema Operativo para el desarrollo del sistema.
	RNFDI 6	Se utilizará Visual Paradigm como herramienta de modelado.
	RNFDI 7	Para el diseño de las páginas se utilizarán las Hojas de Estilo en Cascadas (<i>Cascading Style Sheets, CSS por sus siglas en inglés</i>), Lenguaje de Marcas o Etiquetas de Hipertexto (<i>Hyper Text Markup Language, HTML por sus siglas en inglés</i>), Lenguaje de Marcas Hipertexto Extensible (<i>eXtensible Hypertext Markup Language, XHTML por sus siglas en inglés</i>), JavaScript y JQuery.
Funcionamiento		
Software	RNFFS 1	Para el funcionamiento de los servidores es necesario Windows 7 o superior.
	RNFFS 2	Para el funcionamiento en la PC cliente será necesario Firefox 6 o superior,

Capítulo 2. Propuesta del Sistema

		Google Chrome 16 o superior. Sistema operativo Windows o Mac, capaces de soportar los navegadores mencionados anteriormente.
Hardware	RNFFH 1	El servidor para aplicaciones web deberá tener las siguientes características mínimas: <ul style="list-style-type: none"> • Procesador Intel Pentium Dual Core a 2.0 Ghz, equivalente o superior. • Tarjeta de red o capacidad de conectividad que soporte 1 Gb. • 1 GB de memoria RAM. • Capacidad de 80 GB de disco duro.
	RNFFH 2	El servidor para la base de datos deberá tener las siguientes características mínimas: <ul style="list-style-type: none"> • Procesador Intel Pentium Dual Core a 2.0 Ghz, equivalente o superior. • Tarjeta de red o capacidad de conectividad. • 1 GB de memoria RAM. • Capacidad de 250 GB de disco duro.
	RNFFH 3	Las estaciones de trabajo deberán cumplir los siguientes requisitos mínimos: <ul style="list-style-type: none"> • Procesador Intel Celeron E3200 a 2.40 Ghz, equivalente o superior. • 256 MB de memoria RAM.
Seguridad		
Confidencialidad	RNFSC 1	El acceso al sistema así como la información se encontrarán protegidos contra accesos no autorizados utilizando mecanismos de autenticación propios del sistema.
	RNFSC 2	La autenticación del sistema será la primera acción del usuario, el cual deberá proporcionar un usuario único y una contraseña, los cuales serán de uso exclusivo del propio usuario.
	RNFSC 3	Las diferentes áreas del sistema se encontrarán protegidas contra acceso no autorizado utilizando roles y grupos de usuarios.
	RNFSC 4	El administrador del sistema como política de seguridad podrá restringir el acceso a las diferentes áreas del sistema a los usuarios o grupos de usuarios.
Integridad	RNFSI 1	La información podrá ser modificada solo por el personal autorizado.

Capítulo 2. Propuesta del Sistema

	RNFSI 2	Se harán validaciones de la información en el servidor contra ataques de inyección HTML y SQL.
	RNFSI 3	Se protegerá la información entrada contra ataques CSRF.
Fiabilidad		
	RNFF 1	La respuesta del sistema ante una búsqueda bajo criterios no debe exceder los 5 segundos.
	RNFF 2	Ante una inserción de datos, vista de detalles, modificación o eliminación el sistema debe responder en no más de 3 segundos.
	RNFF 3	El sistema debe responder en un promedio de 6 segundos la petición de un reporte.
	RNFF 4	El sistema deberá soportar transacciones de cerca de 300 usuarios conectados simultáneamente.
	RNFF 5	El sistema podrá ser usado de forma extendida permitiendo un tiempo de mantenimiento cada 6 meses.
	RNFF 6	Ante un fallo o error del sistema se debe restablecer el sistema en un tiempo no mayor de 10 horas.
	RNFF 7	Ninguna información ingresada en el sistema será eliminada físicamente de la base de datos, independientemente de su inexistencia para el sistema. Permitirá la recuperación de la información de la base de datos, a partir de respaldos, o salvvas realizadas.
	RNFF 8	Se realizarán salvvas periódicas de la base de datos para prevenir la pérdida de información.
Eficiencia		
	RNFE 1	El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria.
	RNFE 2	Se respetarán buenas prácticas de programación para incrementar el rendimiento en operaciones costosas.
Interfaz de Usuario		
	RNFIU 1	Las interfaces del sistema contendrán los datos de forma estructurada, permitiendo la interpretación correcta de la información.
	RNFIU 2	La entrada incorrecta de datos será mostrada al usuario claramente, detallando los campos donde se encuentra el error y mostrando como título el detalle del

		error.
	RNFIU 3	Todos los textos y mensajes en pantalla serán mostrados según el idioma seleccionado para el sistema.
	RNFIU 4	El diseño de la interfaz del sistema responderá a la ejecución de acciones de forma rápida, minimizando los pasos a dar en cada proceso.
Interconexión		
	RNFI 1	La PC cliente se comunicará mediante el Protocolo de Transferencia de Hipertexto (HTTP por sus siglas en inglés <i>Hypertext Transfer Protocol</i>) y Protocolo de Transferencia de Hipertexto Seguro (<i>Hypertext Transfer Protocol Secure, HTTPS por sus siglas en inglés</i>) con el servidor de aplicaciones, este se comunicará mediante el Protocolo de Control de Transmisión/Protocolo de Internet (<i>Transmission Control Protocol/ Internet Protocol, TCP/IP por sus siglas en inglés</i>) con el servidor de bases de datos.
	RNFI 2	Los datos de usuarios y demás datos necesarios para el sistema podrán ser extraídos de los servicios web brindados por el UDDI (Directorio de Servicios web de la UCI).

Tabla 3. Lista de reserva del producto. Requisitos no funcionales.

2.4 Fase de Exploración

La metodología de desarrollo XP comienza con su fase de exploración. Durante esta fase se realiza el proceso de identificación de las Historias de Usuario (HU). Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Durante la fase de exploración, se describen las HU, en correspondencia con cada uno de los requisitos funcionales identificados. También se hace una descripción de todos los artefactos generados durante la fase de Exploración.

2.4.1 Roles y Responsabilidades

Usuario: Define pruebas funcionales y puede ocupar cualquiera de los siguientes roles.

Roles que puede ocupar el usuario:

- Técnico: Es el encargado de revisar la llegada de incidencias, responderlas o asignarlas a un especialista. No tiene acceso al módulo de administración.
- Especialista: Es aquel profesional que se encarga de dar respuesta a las incidencias que no pueden ser solucionadas por el técnico o escalarlas a través de la comunicación con el proveedor de los centros de desarrollo. No tiene acceso al módulo de administración.
- Cliente: Escribe las HU y las pruebas funcionales para validar la implementación de dicho módulo.
- Administrador: Es aquel profesional que tiene la responsabilidad de ejecutar, mantener, operar y asegurar el correcto funcionamiento del sistema informático. Además es el único que tiene pleno acceso a dicho sistema.

2.4.2 Historias de usuario

Las HU son la forma en que se especifican en XP los requisitos del sistema. Estas se escriben desde la perspectiva del cliente, aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas. El contenido de estas debe ser concreto y sencillo. El tratamiento de las HU es muy dinámico y flexible, en cualquier momento pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada HU es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas pocas semanas.


Historia de Usuario	
Código: RF 1.	Nombre Historia de Usuario: Autenticar usuarios.
Modificación de Historia de Usuario Número: Ninguna.	
Referencia: Ninguna.	
Programador: Macdemis Liliana Pozo Guevara.	Iteración Asignada: 2.
Prioridad: Muy alta.	Puntos Estimados: 0.6 semanas.
Riesgo en Desarrollo: Alto.	Puntos Reales: 0.6 semanas.
Descripción: El usuario accede al sistema mediante un formulario de acceso, donde introduce su usuario y contraseña. Luego el sistema, muestra sus respectivas funcionalidades de acuerdo al rol que desempeñe.	
Observaciones: Constituye la interfaz inicial del sistema.	
Prototipo de interfaz:	
	

Tabla 4. Historia de Usuario 1

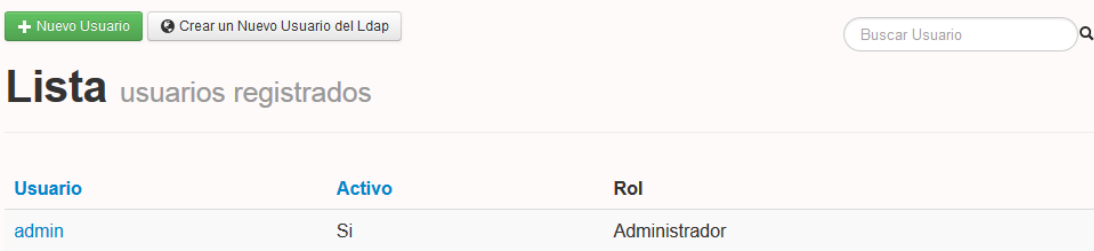
Historia de Usuario	
Código: RF 2.	Nombre Historia de Usuario: Gestionar usuarios.
Modificación de Historia de Usuario Número: Ninguna.	
Referencia: Ninguna.	
Programador: Macdemis Liliana Pozo Guevara.	Iteración Asignada: 2.
Prioridad: Muy alta.	Puntos Estimados: 0.7 semanas.
Riesgo en Desarrollo: Alto.	Puntos Reales: 0.7 semanas.
Descripción: El administrador gestiona los datos del usuario haciendo uso de la base de datos del sistema y accediendo a los datos del LDAP.	
Observaciones: El usuario puede estar activo o inactivo.	
Prototipo de interfaz:	
 <p>The screenshot shows a web interface for user management. At the top, there are two buttons: '+ Nuevo Usuario' (green) and 'Crear un Nuevo Usuario del Ldap' (grey). To the right is a search bar labeled 'Buscar Usuario' with a magnifying glass icon. Below the buttons is the heading 'Lista usuarios registrados'. Underneath is a table with three columns: 'Usuario', 'Activo', and 'Rol'. The table contains one row with the values 'admin', 'Si', and 'Administrador'.</p>	

Tabla 5. Historia de Usuario 2

2.5 Fase de planificación

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondencia, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses.

2.5.1 Estimaciones del esfuerzo

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en

la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

No.	Historia de Usuario	Puntos de Estimación
1	Autenticar usuario	1
2	Gestionar rol	1
2.1	Crear rol	
2.2	Modificar rol	
2.3	Eliminar rol	
3	Gestionar usuario	1
3.1	Crear usuario	
3.2	Eliminar usuario	
3.3	Buscar usuario	
3.4	Modificar usuario.	
4	Gestionar las tareas de los usuarios en el Sistema de Gestión de Servicios de Soporte.	1
4.1	Crear tareas	
4.2	Eliminar tareas	
4.3	Buscar tareas	
4.4	Modificar tareas	
5	Diseñar interfaz del Sistema de Gestión de Servicios de Soporte.	1
6	Diseñar interfaz del Módulo de Administración.	1
7	Diseñar interfaz para Comentario.	1
8	Mostrar datos del Sistema de Gestión de Servicios de Soporte.	1
9	Mostrar ayuda del Sistema de Gestión de Servicios de Soporte.	1

Tabla 6. Plan de esfuerzo por Historia.

2.5.2 Plan de duración de iteraciones

Las historias de usuario escogidas para cada entrega, deben ser desarrolladas y probadas, teniendo en cuenta el ciclo de iteración, de acuerdo con el orden dado a las mismas. Este plan define qué historia de usuario será implementada para cada iteración y las fechas en que serán liberadas. A continuación se

muestran tres iteraciones y el número de las historias de usuario que se encuentran dentro de cada iteración.

Iteraciones	Historia de Usuario	Duración Total
1	<ul style="list-style-type: none">• Diseñar interfaz del Sistema de Gestión de Servicios de Soporte.• Diseñar interfaz del Módulo de Administración.• Diseñar interfaz de la funcionalidad Comentarios.	2 semanas
2	<ul style="list-style-type: none">• Autenticar usuarios• Gestionar rol<ul style="list-style-type: none">Crear rolModificar rolEliminar rol• Gestionar usuarios<ul style="list-style-type: none">Crear usuariosEliminar usuariosBuscar usuariosModificar usuarios	3 semanas
3	<ul style="list-style-type: none">• Mostrar datos del Sistema de Gestión de Servicios de Soporte.• Mostrar ayuda del Sistema de Gestión de Servicios de Soporte.	1 semana

Tabla 7. Plan de duración de las iteraciones.

2.5.3 Plan de entrega

Una vez que el cliente culmina la elaboración de las HU, se comienza con la creación del plan de entregas. El mismo se hace con la intención de que los programadores obtengan una estimación de dichas historias en cuanto al nivel de detalle, o sea, para fijar el período de tiempo que se puede tardar en la implementación de cada una teniendo en cuenta que la implementación empezó en la tercera semana de marzo (15 de marzo).

Artefacto	Iteración	Entrega
RF 5 RF 6 RF 7	Finales de la primera iteración.	27/03/2014
RF 1 RF 2 RF 2.1 RF 2.2 RF 2.3 RF 2.4 RF 3 RF 3.1 RF 3.2 RF 3.3 RF 4 RF 4.1 RF 4.2 RF 4.3 RF 4.4	Finales de la segunda iteración.	15/04/2014
RF 8 RF 9	Finales de la tercera iteración.	21/04/2014

Tabla 8. Plan de entrega de las iteraciones.

2.6 Diseño del sistema

La metodología XP describe el sistema utilizando las tarjetas CRC (Contenido, Responsabilidad y Colaboración). Esto no implica que no se utilicen los diagramas para obtener una mejor visión y comunicación entre el equipo de trabajo, siempre y cuando su complejidad no sea alta y defina información importante.

2.6.1 Tarjetas CRC

Las tarjetas CRC permiten ver las clases no como un depósito de datos, sino para conocer el comportamiento de cada una en un alto nivel. La metodología XP estipula el uso de la misma como un artefacto obligatorio durante el desarrollo de un proyecto, debido a los beneficios que aportan a los desarrolladores. A continuación se muestran las tarjetas de dos de las clases que responden a funcionalidades de alta prioridad en el sistema, el resto puede visualizarse en los anexos.

Tarjeta CRC	
Clase: Comentario	
Responsabilidades	Colaboraciones
<p>Es la encargada de permitirle al usuario autenticado enviarle mensajes de notificación al administrador del sistema. Esta almacena los siguientes atributos:</p> <ul style="list-style-type: none"> ➤ String nombre; ➤ String correo; ➤ String asunto; ➤ String cuerpo; 	Mail Service

Tabla 9. Tarjeta CRC Comentario.

Tarjeta CRC	
Clase: Tareas	
Responsabilidades	Colaboraciones
<p>Es la encargada de almacenar los datos de las tareas asignadas a los usuarios que tienen acceso al Sistema de Gestión de Servicios de Soporte. Esta almacena los siguientes atributos:</p> <ul style="list-style-type: none"> ➤ String titulo; ➤ String descripcion; ➤ Date horainicioProg; ➤ Date horafinProg; ➤ Date horainicioReal; ➤ Date horafinReal; ➤ String comentario; ➤ Usuario tecnico; ➤ String estado; 	Usuario

Tabla 10. Tarjeta CRC Tareas.

2.6.2 Patrones de diseño

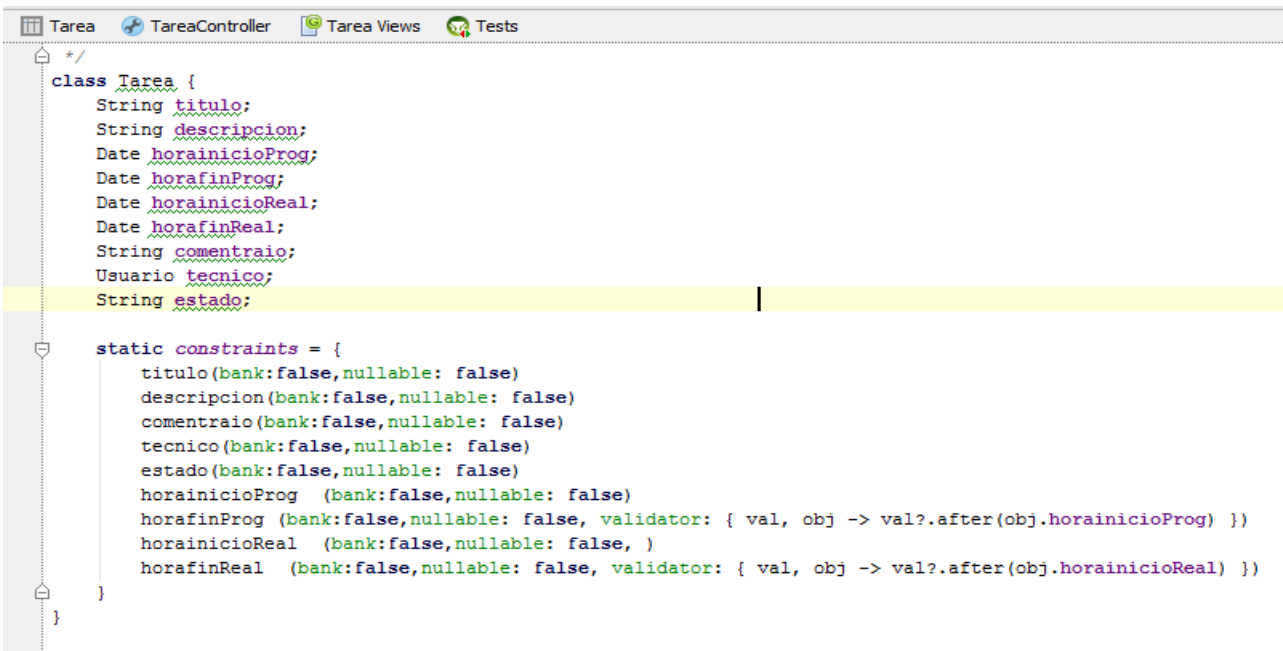
El uso de patrones proporciona una estructura conocida por todos los programadores, de manera que la forma de trabajar no resulte distinta entre los mismos. GRASP⁴ constituye una serie de buenas prácticas de aplicación recomendable en el diseño de software. Entre ellos se encuentran el experto, creador, alta cohesión, bajo acoplamiento y el controlador.

Experto

El GRASP de experto en información es el principio básico de asignación de responsabilidades. Nos indica, por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo obtendremos un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento) (ERICH GAMMA, 2013).

A continuación se muestra un ejemplo del uso del patrón experto en la clase Tarea del sistema. La clase Tarea permite establecer la relación existente entre cada elemento por la cual está compuesta para así poder gestionar la misma. La figura 6 Ejemplo del uso del patrón Experto muestra la estructura que presenta dicha clase.

⁴ GRASP: Patrones Generales de Software para Asignar Responsabilidades.



```
class Tarea {
    String titulo;
    String descripcion;
    Date horainicioProg;
    Date horafinProg;
    Date horainicioReal;
    Date horafinReal;
    String comentaio;
    Usuario tecnico;
    String estado;

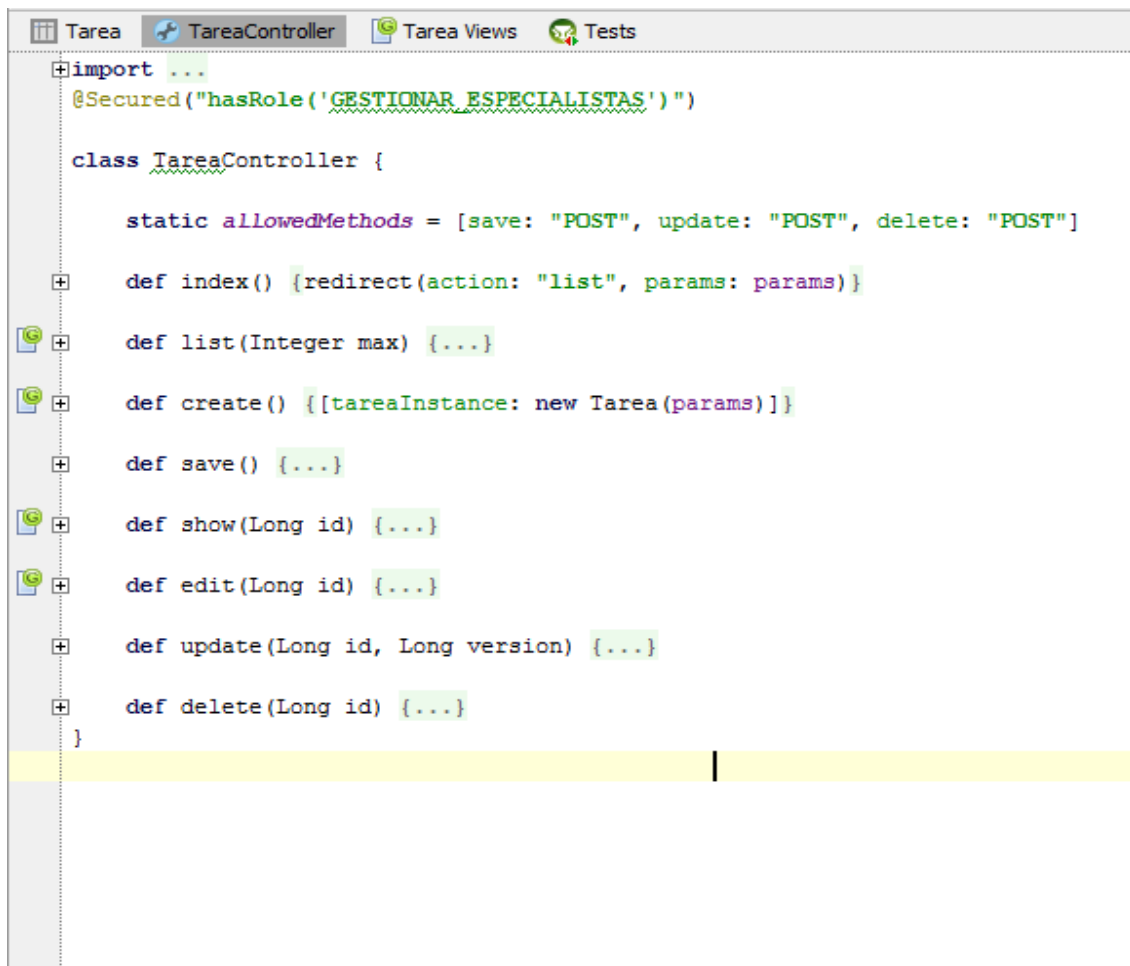
    static constraints = {
        titulo(bank:false,nullable: false)
        descripcion(bank:false,nullable: false)
        comentaio(bank:false,nullable: false)
        tecnico(bank:false,nullable: false)
        estado(bank:false,nullable: false)
        horainicioProg (bank:false,nullable: false)
        horafinProg (bank:false,nullable: false, validator: { val, obj -> val?.after(obj.horainicioProg) })
        horainicioReal (bank:false,nullable: false, )
        horafinReal (bank:false,nullable: false, validator: { val, obj -> val?.after(obj.horainicioReal) })
    }
}
```

Figura 6. Ejemplo del uso del patrón Experto.

Creador

El patrón creador nos ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. El patrón creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos.

La clase TareaController evidencia el uso del patrón creador ya que es la encargada de crear, modificar, listar y eliminar las instancias de la clase Tarea. La figura 7 Ejemplo del uso del patrón creador se evidencia su uso en la estructura de dicha clase.



```
import ...
@Secured("hasRole('GESTIONAR_ESPECIALISTAS')")

class TareaController {

  static allowedMethods = [save: "POST", update: "POST", delete: "POST"]

  def index() {redirect(action: "list", params: params)}

  def list(Integer max) {...}

  def create() {[tareaInstance: new Tarea(params)]}

  def save() {...}

  def show(Long id) {...}

  def edit(Long id) {...}

  def update(Long id, Long version) {...}

  def delete(Long id) {...}
}
```

Figura 7. Ejemplo del uso del patrón creador.

Bajo acoplamiento

Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

Alta Cohesión

La información que almacena una clase debe de ser coherente y está en la mayor medida de lo posible relacionada con la clase. La cohesión es una medida de cuán relacionadas y enfocadas están las

responsabilidades de una clase. Una clase con alta cohesión, compartirá la responsabilidad de una operación, con otras clases.

Controlador

El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control.

2.6.3 Patrones de arquitectura

Los patrones de arquitectura expresan un esquema organizativo estructural fundamental para sistemas de software. En el sistema se está trabajando con Grails, este framework está basado en el patrón MVC⁵, por lo que se hará uso del mismo para facilitar la realización del sistema a desarrollar.

El patrón de arquitectura MVC permite realizar la programación multicapa, separando en tres componentes distintos como se muestra en la figura 8 Patrón Modelo Vista Controlador, los datos de una aplicación, la interfaz del usuario y la lógica de control. 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 la lógica de negocio; el controlador es el responsable de recibir los eventos de entrada desde la vista.

⁵ MVC: Modelo Vista Controlador.

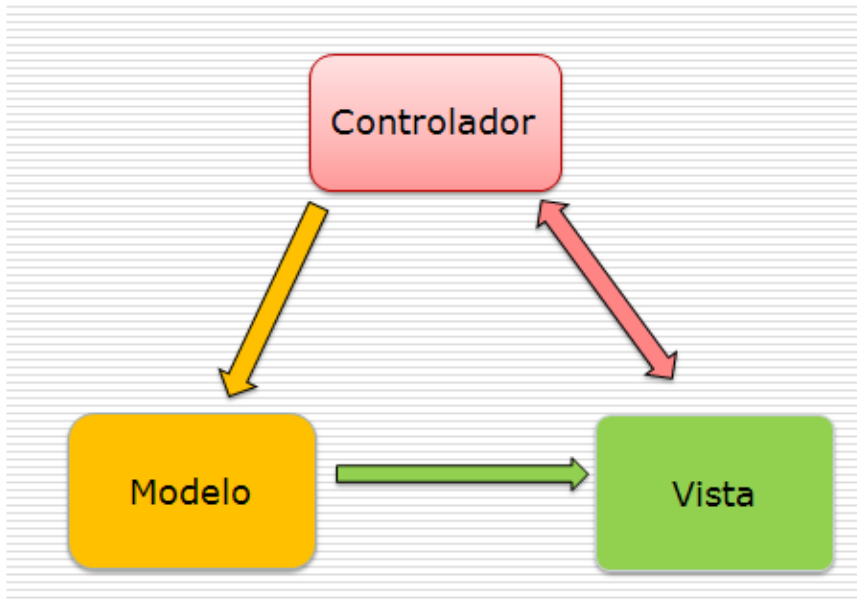


Figura 8. Patrón Modelo Vista Controlador.

El ciclo de vida de este patrón consta de tres etapas. La primera, es la que inicia el usuario cuando realiza una petición a la aplicación, el controlador recibe la solicitud hecha por el cliente y decide a quién debe delegar esa tarea.

Modelo: Representa la información con la que trabaja la aplicación, o sea, su lógica de negocio. La figura 9 Ejemplo del componente modelo en el sistema muestra la estructura de la clase Comentario como ejemplo de modelo existente en el sistema.

```
Comentario ComentarioController Comentario Views Tests
+/**...*/
class Comentario {
    String nombre
    String correo
    String asunto
    String cuerpo

    static constraints = {
        nombre(size: 1..999)
        correo(email:true)
        asunto(size:1..50)
        cuerpo(size:0..999999)
    }
}
```

Figura 9. Ejemplo del componente modelo en el sistema.

Vista: Convierte el modelo en una página web que facilita al usuario interactuar con ella. En la Figura 10 Ejemplo del componente vista en el sistema desde la web se muestra el resultado de la vista correspondiente a la clase Comentario.



The screenshot shows a web application interface for a support center. The header features the text "Centro de Soporte" in a stylized font, accompanied by a background image of a person at a computer and a woman wearing a headset. Below the header is a navigation menu with items: "Panel de Control", "Solicitudes", "Problemas", "Cambios", "Soluciones", "Contratos", and "Bienvenido admin". A secondary menu below that includes "Admin." and "Soporte". The main content area is titled "Enviar Comentario" and contains a form with the following fields: "Para: mlpozo@estudiantes.uci.cu", "Su nombre:", "Su correo electrónico:", "Asunto:", and "Mensaje:". A "Crear" button is located below the message field. The footer of the page reads "Universidad de las Ciencias Informáticas".

Figura 10. Ejemplo del componente vista en el sistema.

Controlador: Es el encargado de procesar las interacciones del usuario y ejecuta los cambios adecuados en el modelo o en la vista. En la figura 11 Ejemplo del componente controlador en el sistema se muestra la estructura de la clase ComentarioController en el sistema.

```
Comentario ComentarioController Comentario Views Tests
class ComentarioController {
    static allowedMethods = [save: "POST"]

    def index() {
        redirect(action: "create", params: params)
    }

    def create() {
        [comentarioInstance: new Comentario(params)]
    }

    def save() {
        def comentarioInstance = new Comentario(params)

        sendMail {
            to "'Macdemis Liliana Pozo Guevara' <mlpozo@estudiantes.uci.cu>"
            from "'Macdemis Liliana Pozo Guevara' <mlpozo@estudiantes.uci.cu>"
            cc comentarioInstance.correo
            subject comentarioInstance.asunto
            html comentarioInstance.nombre+":\n"+
                comentarioInstance.cuerpo
        }
        redirect(action: "create")
        return
    }
}
```

Figura 11. Ejemplo del componente controlador en el sistema.

2.7 Modelo Conceptual

El modelo conceptual también conocido como modelo de dominio, permite definir los conceptos relevantes y su relación en la descripción de un problema o en un dominio de interés en el mundo real. Proporcionando de esta forma una fácil comprensión de la perspectiva conceptual del negocio y disminuyendo la brecha de representación de como ve el cliente el problema y la representación en

software de la solución. Con este modelo no se deben mostrar o representar componentes de software como clases y operaciones.

En el Módulo de Administración para el Sistema de Gestión de Servicios de Soporte se llevó a cabo el desarrollo de un modelo conceptual. En este se visualizan las clases conceptuales y las asociaciones entre ellas, implicadas en el Sistema de Gestión de Servicios de Soporte y el Módulo de Administración para dicho sistema. El Sistema de Gestión de Servicios de Soporte muestra en la interfaz principal un resumen con las tareas asignadas según el usuario loggeado, en caso de este poseer rol de Administrador o Especialista va a poder agregar una nueva tarea o ver el listado de todas las que están registradas en el sistema; sin embargo, si su rol es de técnico, solo va a poder visualizar sus tareas y actualizar el estado de las misma. Proporcionando además un listado de avisos que va a ser público para todos los usuarios, donde estos podrán agregar uno nuevo, eliminarlo, modificarlo o ver los detalles del mismo. El sistema permite enviar comentarios a la administración del sitio sobre alguna inquietud o incidencia; así como también va existir la posibilidad de acceder a la ayuda donde se va poder aprender cómo hacer uso del mismo. Esta aplicación web consta de un Módulo de Administración, donde en el diseño de la interfaz principal se van a encontrar los accesos para la administración de los diferentes módulos que se van a integrar posteriormente al Sistema de Gestión de Servicios de Soporte, así como una breve descripción de la misión de estos. Dentro de estos accesos se va a poder acceder a la funcionalidad gestionar usuarios, donde el administrador del sistema va a poder adicionar, eliminar, modificar y ver el listado de todos los usuarios registrados con sus respectivo rol; así como también gestionar los roles especificados en el sistema.

Administrador: Es el usuario con más alto nivel de privilegio en el sistema, encargado de monitorear y controlar los procesos realizados en la aplicación.

Especialista: Es aquel profesional que se encarga de dar respuesta a las incidencias que no pueden ser solucionadas por el técnico o escalarlas a través de la comunicación con el proveedor de los centros de desarrollo. No tiene acceso al módulo de administración.

Técnico: Es el encargado de revisar la llegada de incidencias, responderlas o asignarlas a un especialista. No tiene acceso al módulo de administración.

Usuario: Entidad que identifica a las personas con acceso al sistema.

Rol: Define los diferentes niveles de permisos en el sistema.

2.7 Diseño de la base de datos

El modelado del diseño físico de una base de datos, consiste en producir una descripción de la implementación de la base de datos. Esta descripción incluye las estructuras de almacenamiento y los métodos de acceso que se utilizarán para conseguir un acceso eficiente a los datos.

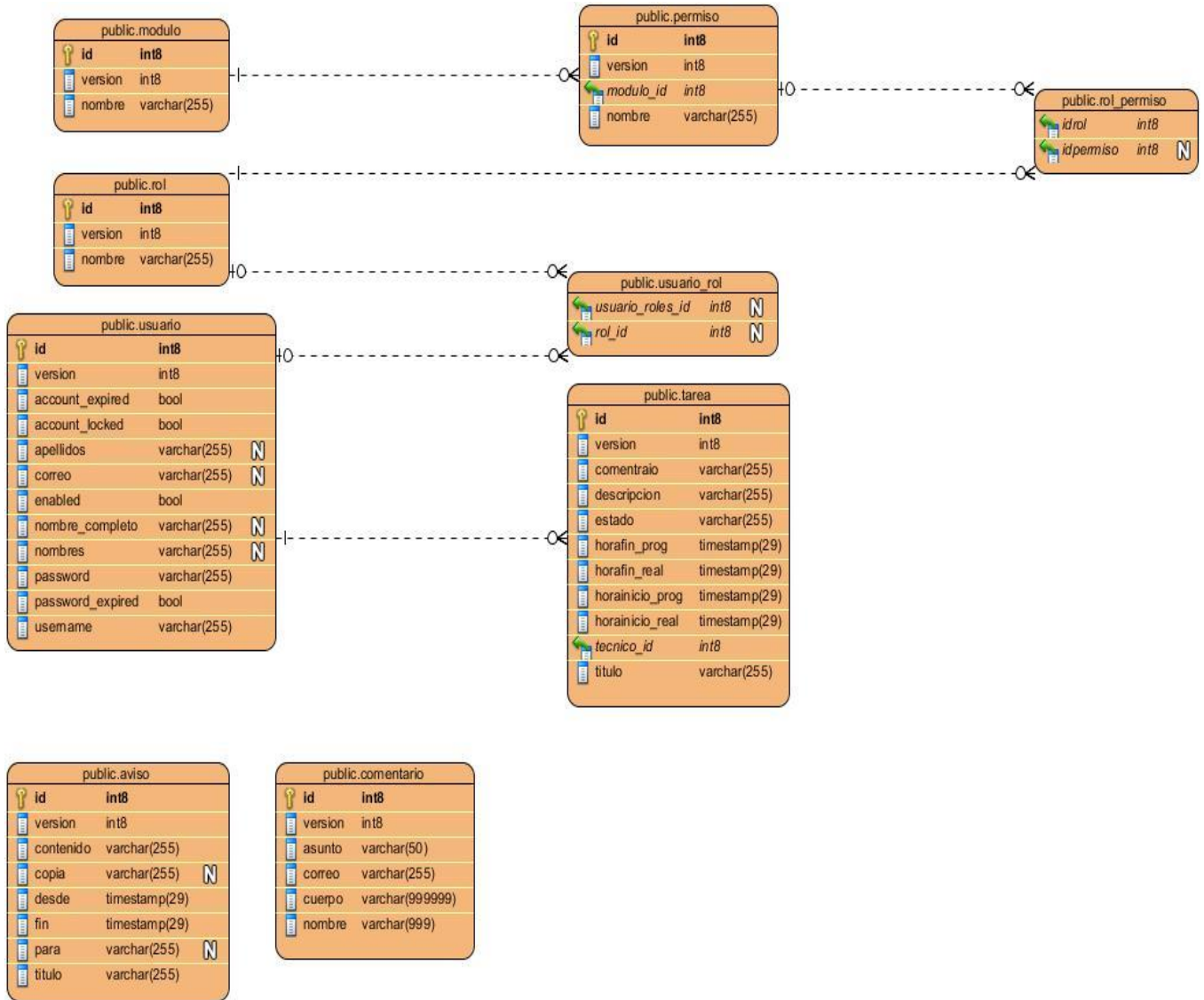


Figura 13. Modelo Físico.

En la figura anterior se muestra el modelo físico de la base de datos utilizada en el desarrollo de la aplicación. Esta cuenta con 9 entidades, siendo las entidades de Usuarios y Tareas las de mayor relevancia. Esta relevancia viene dada porque constituyen el pilar principal del flujo de trabajo del sistema. Las entidades restantes definen los diferentes niveles de accesibilidad y usabilidad; así como también actividades complementarias que facilitan la interacción con el sistema.

2.9 Fase de implementación

Después de terminado el diseño se procede a desarrollar la solución teniendo en cuenta los estándares de codificación.

2.9.1 Estándares de codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Estos estándares facilitan el mantenimiento del código, sirven como punto de referencia para los programadores, mantienen un estilo de programación y ayudan a mejorar el proceso de codificación, haciéndolo, entre otras cosas, mucho más eficiente.

En la implementación del sistema se va a utilizar la notación CamelCase. Groovy es un lenguaje de programación basado en java, en el lenguaje java, se usa la notación CamelCase en identificadores para clases, y dromedaryCase para métodos y variables. La notación Camel consiste en escribir los identificadores con la primera letra de cada palabra en mayúsculas y el resto en minúscula. Se llama notación "Camel" porque los identificadores recuerdan las jorobas de un camello. Existen dos variantes:

- UpperCamelCase, CamelCase o PascalCase: en esta variante la primera letra también es mayúscula.
- lowerCamelCase, camelCase o dromedaryCase: la primera letra es minúscula.

Identación

Se utiliza el estilo Allman, también conocido como ANSI style. Este estilo ubica las llaves asociadas a una sentencia de control en la siguiente línea, las sentencias son indentadas al siguiente nivel de las llaves con 1 o 2 TAB de espacio. Ejemplo:

```
If (Vacio())  
{  
Operacion1 ();  
Operacion2 ();  
}
```


2.9.2 Implementación

Durante el transcurso de las iteraciones se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Las HU agrupadas en cada iteración se van implementando durante el transcurso de la iteración a la cual pertenecen. Al principio de cada iteración se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan se descomponen las HU en tareas de desarrollo. Estas tareas son para el uso estricto del programador o los programadores, por lo que pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente.

A continuación quedan detalladas las tareas de desarrollo realizadas en cada una de las iteraciones.

Iteración 1

En esta iteración se da cumplimiento a las HU que corresponden a los números 5, 6 y 7 las cuales comprenden la realización del diseño de las interfaces del sistema a desarrollar.

Historias de Usuario	Tiempo de implementación	
	Estimación	Real
Diseñar interfaz del Sistema de Gestión de Servicios de Soporte.	1	0.6 semanas.
Diseñar interfaz del Módulo de Administración.	1	0.7 semanas.
Diseñar interfaz de la funcionalidad Comentarios.	1	0.4 semanas.

Tabla 11. HU abordadas en la iteración I

Tareas de las historias de usuarios desarrolladas en la primera iteración

Luego de relacionar las HU pertenecientes a esta iteración, se procede a la especificación de las principales tareas de desarrollo, que se realizaron para cumplir el propósito de la misma. A continuación se muestra la tarea de desarrollo correspondiente a la HU 5, las tareas de las restantes HU pueden evidenciarse en los anexos.

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: HU_5
Nombre Tarea: Diseñar interfaz del Módulo de Administración para el Sistema de Gestión de Servicios de Soporte.	
Tipo de Tarea: Diseño.	Puntos Estimados: 0.7 semanas
Fecha Inicio: 19/03/2013	Fecha Fin: 23/03/2013
Arquitecto del Sistema: Macdemis Liliana Pozo Guevara.	
Descripción: Desarrollar la interfaz principal del Módulo de Administración para el Sistema de Gestión de Servicios de Soporte, según los requisitos especificados por el cliente.	

Tabla 12. Descripción de la tarea de desarrollo #2

Iteración 2

En esta iteración se implementan los HU correspondientes a los números 1, 2, 2.1, 2.2, 2.3, 2.4, 3, 3.1, 3.2, 3.3, 4, 4.1, 4.2, 4.3, 4.4 las cuales permiten la autenticación, gestión de usuarios, roles y tareas en el sistema;

Historias de Usuario	Tiempo de implementación	
	Estimación	Real
Autenticar usuario.	1	0.6 semanas.
Gestionar usuarios.	1	0.7 semanas.
Crear usuarios.	1	0.3 semanas.
Eliminar usuarios.	1	0.1 semanas.
Buscar usuarios.	1	0.1 semanas.
Modificar usuarios.	1	0.1 semanas.
Gestionar rol	1	0.7 semanas.
Crear rol	1	0.3 semanas.

Modificar rol	1	0.3 semanas.
Eliminar rol	1	0.1 semanas.
Gestionar las tareas de los usuarios en el Sistema de Gestión de Servicios de Soporte.	1	0.7 semanas.
Crear tarea.	1	0.3 semanas.
Eliminar tarea.	1	0.1 semanas.
Buscar tarea.	1	0.1 semanas.
Modificar tarea.	1	0.1 semanas.

Tabla 13. HU abordadas en la iteración 2

Tareas de las historias de usuarios desarrolladas en la segunda iteración

A continuación se muestra la tarea de desarrollo perteneciente a la HU 1.

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: HU_1
Nombre Tarea: Implementación de la funcionalidad Autenticar usuarios.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.6 semanas.
Fecha Inicio: 23/03/2014.	Fecha Fin: 27/03/2014.
Programador Responsable: Macdemis Liliana Pozo Guevara.	
Descripción: Implementar la funcionalidad para que el usuario acceda al Sistema de Gestión de Servicios de Soporte mediante un formulario de acceso.	

Tabla 14. Descripción de la tarea de desarrollo #1

Iteración 3

En esta última iteración se implementó las HU correspondientes a los números 8 y 9, las cuales permiten generar los datos seleccionados por el usuario en el sistema.

Historias de Usuario	Tiempo de implementación	
	Estimación	Real
Mostrar ayuda del Sistema de Gestión de Servicios de Soporte.	1	0.4 semanas.
Mostrar datos del Sistema de Gestión de Servicios de Soporte.	1	0.4 semanas.

Tabla 15. HU abordadas en la iteración 3

Tareas de las historias de usuarios desarrolladas en la tercera iteración

A continuación se muestra la tarea de desarrollo perteneciente a la HU 8.

Tarea de Ingeniería	
Número Tarea: 8	Número Historia de Usuario: HU_8
Nombre Tarea: Implementación de la funcionalidad Mostrar datos del Sistema de Gestión de Servicios de Soporte.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 0.4 semanas
Fecha Inicio: 19/04/2014	Fecha Fin: 21/04/2014
Programador Responsable: Macdemis Liliana Pozo Guevara.	
Descripción: Implementar la funcionalidad para que se muestren los datos del Sistema de Gestión de Servicios de Soporte.	

Tabla 16. Descripción de la tarea de desarrollo #8

2.10 Conclusiones parciales

La metodología de desarrollo seleccionada para guiar el proceso de software, posibilitó la descripción de los roles que interactuarán con el sistema, para garantizar la seguridad de acceso al contenido. En conjunto con el cliente se definieron las HU para la especificación funcional, facilitando la comunicación del programador con el mismo a la hora de implementarlas, teniendo en cuenta el plan de iteraciones.

Se estimó el tiempo que tomará llevar a cabo el desarrollo de la misma considerándose que debe estar listo en el tiempo establecido. Además, se especificó el diseño de clases del sistema mediante la utilización de las tarjetas CRC. La descripción de los patrones de diseño y estándares de codificación permitieron legibilidad en el código y una mayor comprensión del mismo. Las iteraciones de desarrollo sobre el sistema, permitieron que al finalizar, se obtuviera un producto con todas las restricciones y características deseadas por el cliente. De esta forma se da paso a la construcción de la solución para luego validar los resultados que se obtendrán mediante las pruebas.

Capítulo 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

3.1 Introducción

En el presente capítulo se describen las pruebas realizadas al Módulo de Administración para el Sistema de Gestión de Servicios de Soporte del Centro de Soporte UCI.

3.2 Pruebas

Uno de los pilares de la metodología XP es el uso de las pruebas para comprobar el funcionamiento de los códigos que se vayan implementando.

3.2.1 Pruebas unitarias

Uno de los métodos más utilizados para realizar pruebas a nuestro software es el llamado pruebas unitarias (*unit testing*). La base de este método es el hacer pruebas en pequeños fragmentos a nuestro sistema, o sea verificar que cada método y función esté trabajando correctamente. Cada prueba deberá ser lo más independientemente posible de las demás.

A continuación se ejemplifica una de las pruebas unitarias realizadas al sistema.

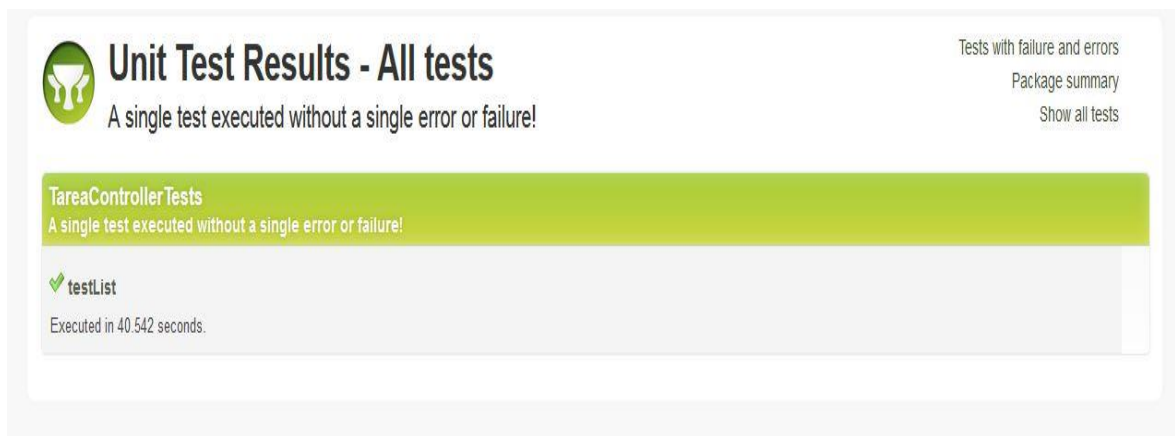


Figura 14. Prueba unitaria realizada al método List de la clase TareaController.

Análisis de resultados

Con la realización de las pruebas unitarias al sistema se pudo comprobar que los métodos se comportan de forma correcta, sin tener en cuenta su entorno. Esto significa que cuando se ejecutan las pruebas

unitarias de un método Grails no inyectará ninguno de los métodos dinámicos con los que contamos cuando la aplicación se está ejecutando.

3.2.2 Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las HU, en cada ciclo de iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una HU ha sido correctamente implementada. Las pruebas de aceptación son consideradas como “pruebas de caja negra”. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación.

Para cada una de las pruebas de aceptación realizadas en el sistema se obtuvo una evaluación, de acorde al resultado de la prueba realizada se emitirá una evaluación sobre la misma. Esta evaluación tendrá uno de los tres resultados que a continuación se describen:

- **Bien:** cuando el resultado de la prueba es exactamente el esperado por el usuario.
- **Parcialmente bien:** cuando el resultado no es completamente el esperado por el cliente o usuario de la aplicación y muestra resultados erróneos o fuera de contexto.
- **Mal:** cuando el resultado de la prueba realizada genera un error de codificación en la aplicación o muestra como resultado elementos no deseados o fuera de contexto, trayendo como consecuencia que la funcionalidad requerida por el cliente no tenga resultado, lo que invalida también la HU.

A continuación se ejemplifica uno de los casos de pruebas de aceptación realizados a las historias de usuario.

Caso de prueba de aceptación	
Código: HU 1_P1	Historia de usuario: 1
Nombre: Autenticar usuario.	
Descripción: El usuario realiza la autenticación al sistema.	
Condiciones de ejecución: El usuario y la contraseña tienen que ser válidos.	
Entradas/Pasos de ejecución:	

El usuario accede al sistema introduciendo su usuario y contraseña en un formulario de acceso.
El sistema, una vez logueado el usuario, muestra la vista principal del sistema y según el rol que desempeña el mismo, las funcionalidades a las que tiene acceso.
Resultado de la prueba: El sistema muestra la vista principal del sistema.
Evaluación de la prueba: Bien.

Tabla 17. Prueba de aceptación de la historia de usuario 1

Análisis de resultados

Se realizaron 13 pruebas funcionales para validar que el sistema funcionara correctamente, mostrando las salidas correspondientes a cada escenario. De estas pruebas realizadas en una primera iteración fueron clasificadas de bien 5 pruebas de 5 pruebas realizadas. En una segunda iteración de un total de 5 pruebas realizadas, fue clasificada de bien 5 de ellas. En una tercera y última iteración de un total de 3 pruebas realizadas, resultaron todas satisfactorias, constituyendo un 100% de pruebas funcionales exitosas.

3.2.3 Pruebas de Carga

Las pruebas de carga fueron realizadas al sistema para determinar el tiempo de respuesta del servidor web ante peticiones concurrentes hechas por los usuarios.

Dando como resultado el tiempo Promedio de respuesta y el tiempo Máximo que se puede demorar el sistema ante peticiones concurrentes hechas por 200 usuarios. Esta prueba se realizó con la herramienta de carga JMeter que permite llevar a cabo simulaciones sobre cualquier recurso de software. Posee la capacidad de realizar desde una solicitud sencilla hasta secuencias de mayor envergadura simulando la aplicación en tiempo de ejecución. Para la realización de la prueba fue usada una computadora servidor con un microprocesador AMD-E300 (frecuencia: 1.3GHz), memoria RAM de 4GB y disco duro de 320GB y como cliente fue usada una computadora con microprocesador AMD-E300 (frecuencia: 1.3GHz), memoria RAM de 4GB y disco duro de 320GB conectadas a través de tarjetas de red Fast Ethernet con velocidad de 100Mbps.

En la figura 15 Parámetros de rendimientos evaluados por la herramienta JMeter se muestran los parámetros evaluados por dicha herramienta:

Capítulo 3. Validación de la Solución Propuesta

- Página a la que se accede
- Número de muestras
- Tiempo medio en milisegundos
- Mediana en milisegundos
- Línea de 90% en milisegundos
- Tiempo mínimo en milisegundos
- Tiempo máximo en milisegundos
- Porcentaje de Error
- Rendimiento
- Kb/segundo

Etiqueta	Media	# Muestras	Mediana	Línea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/Nuevo/	20809	200	17768	35004	1753	37736	0,00%	5,1/sec	19,7
/Nuevo/login/autenticarUsuario	35069	200	34882	56435	3632	67806	0,00%	2,1/sec	8,3
/Nuevo/images/Fondos/Blue_fireflies....	1094	200	289	3257	17	7420	0,00%	2,1/sec	278,8
/Nuevo/controladora/admin	13766	200	11591	24433	1719	30657	0,00%	1,7/sec	7,0
/Nuevo/usuario/index	13864	200	12218	23200	1098	35148	0,00%	1,6/sec	6,4
/Nuevo/usuario/registroLdap	13533	600	12246	23438	247	36027	0,00%	3,7/sec	14,2
/Nuevo/usuario/buscarLdapUser	11757	400	10134	21798	230	36416	0,00%	2,8/sec	10,7
/Nuevo/usuario/registroLdap...	9820	200	8096	19295	215	35291	0,00%	1,5/sec	5,9
Total	14412	2200	11459	31425	17	67806	0,00%	11,8/sec	184,3

Figura 15. Parámetros de rendimiento evaluados por la herramienta JMeter.

A partir de los valores obtenidos en la tabla, se llega a la conclusión que la aplicación tuvo el comportamiento esperado en las condiciones en que se realizó la prueba. Se obtuvieron tiempos de rendimiento satisfactorios para 200 peticiones concurrentes utilizando los equipos descritos anteriormente.

3.3 Conclusiones parciales

Con la realización de las pruebas unitarias se demostró que el funcionamiento de los métodos cumple con el comportamiento esperado, lo que agiliza el proceso de integración. Las pruebas de aceptación demostraron que el sistema funciona correctamente y con las de carga y estrés se demostró lo mismo obteniendo satisfactorios tiempos de rendimiento. Por lo que se ha arribado a la conclusión que mediante todas las pruebas realizadas a la aplicación, el sistema cumple de manera exitosa con las funcionalidades identificadas durante el proceso de desarrollo del software llevado a cabo en la presente investigación.

Conclusiones generales

Con el propósito de darle cumplimiento al objetivo general y a la problemática planteada en el presente trabajo, se han llevado a cabo satisfactoriamente cada una de las tareas que fueron trazadas al comienzo del mismo.

- Con el estudio de los procesos de la Gestión de Servicios, se definió el marco teórico conceptual de la investigación, que permitió identificar la situación problemática.
- El estudio de los sistemas homólogos demostró la carencia de un sistema que cumpla con las necesidades del Centro de Soporte, por lo que fue necesario el desarrollo de las Interfaces del Módulo de Administración. Además dicho estudio proporcionó un conjunto de características que fueron incorporadas a la solución propuesta.
- La utilización de la metodología ITIL facilitó el desarrollo de las Interfaces del Módulo de Administración, pues define el flujo de trabajo de los procesos de gestión de servicios del área El Soporte de Servicio, lo cual sirvió de guía para la implementación.
- Se logró la generación de los artefactos propuestos por la metodología seleccionada para el desarrollo de la solución, con lo cual quedaron definidos los requerimientos exigidos por el cliente.
- Fue probado el sistema, lo que demostró el cumplimiento de las exigencias del cliente.
- El desarrollo de las Interfaces para el Módulo de Administración contribuye a la futura integración y gestión de los módulos de Gestión de Incidencias, Problemas, Cambios, Disponibilidad, Catálogo, Niveles de Servicios y la Base de Datos de la Gestión de Configuraciones (*CMDB*, por sus siglas en inglés) y los procesos involucrados en el Sistema de Gestión de Servicios del centro de soporte de la UCI respectivamente.

Recomendaciones

Con vistas a mejorar el trabajo se recomienda:

- Integrar los módulos de Gestión de Incidencias, Problemas, Cambios, Disponibilidad, Catálogo, Niveles de Servicios y la Base de Datos de la Gestión de Configuraciones (*CMDDB*, por sus siglas en inglés) a las Interfaces del Módulo de Administración para el Sistema de Gestión de Servicios del Centro de Soporte UCI.
- Realizar la correcta gestión de los procesos involucrados en el Sistema de Gestión de Servicios de Soporte a través del completo desarrollo del Módulo de Administración.

Referencias Bibliográficas

- 1-Advantare. 2014. Advantare Consultores. [En línea] 2014. <http://www.avantare.com>.
- 2-Coello, Helkyn. 2008. Helkyn Coello Blog. Información de interés del mundo TI. . *ITIL, COBIT, CMMI, PMBOK: Como integrar y adoptar los estándares para un buen Gobierno de TI*. [En línea] 8 de 10 de 2008. [Citado el: 7 de 11 de 2013.] <http://helkyncoello.wordpress.com/2008/12/08/itil-cobit-cmmi-pmbok-como-integrar-y-adoptar-los-estandares-para-un-buen-gobierno-de-ti/>.
- 3-Ingenius. 2009. E.T.S de Ingeniería informática- Universidad de Sevilla. *E.T.S de Ingeniería informática- Universidad de Sevilla*. [En línea] 2009. [Citado el: 5 de 11 de 2013.] <http://ingenius.informatica.us.es/index.php/descargas/gestiondeserviciosti>.
- 4-Harold Koontz, Heinz Weirich. 2004. *Administración Un Perspectiva Global*. s.l. : McGraw-Hill Interamericana, 2004.
- 5-CA technologies. 2012. *CA Service Desk Manager*. 2012.
- 6-Customer Care Associates. 2011. [En línea] 2011. http://www.customercareassoc.com/index.php?option=com_content&view=article&id=190%3Aoutsourcing-service-desk-una-opcion-para-alinearnos-al-negocio&catid=21%3Ageneral&Itemid=1.
- 7-Ireo. 2002. Ireo. [En línea] 2002. [Citado el: 13 de 11 de 2013.] [URL:http://www.ireo.com/fabricantes-y-productos/manageengine/servicedesk-plus/resumen/](http://www.ireo.com/fabricantes-y-productos/manageengine/servicedesk-plus/resumen/).
- 8-Riverón, Yaíma Oval. 2007. *Servicio de Soporte Técnico utilizando la tecnología Service – Desk*. S.l: UCI, 2007.
- 9-Silva, Reinaldo Oliveira Da. 2002. *Teorías de la Administración*. s.l. : International Thomson Editores, S.A. de C.V., 2002.
- 10-Stephen Robbins, Mary Coulter. 2005. *Administración*. s.l. : Pearson Educación, 2005.
- 11-Sun Microsystems. 2003. ITIL® IT Infrastructure Library. *Las Mejores Prácticas para la Gestión de Servicios de TI en su Organización*. [En línea] 2003. [Citado el: 10 de 11 de 2013.] sun.es/services/itil.
- 12-Totem. [En línea] <http://www.totemguard.com/netsupport-servicedesk/gestion-incidencias-helpdesk-preguntas-frecuentes/#P2>.

- 13-CA technologies. 2012. *CA Service Desk Manager*. 2012.Ireo. 2002. Ireo. [En línea] 2002. [Citado el: 13 de 11 de 2013.] [URL:http://www.ireo.com/fabricantes-y-productos/manageengine/servicedesk-plus/resumen/](http://www.ireo.com/fabricantes-y-productos/manageengine/servicedesk-plus/resumen/).
- 14-Scribd. 2014. Metodologías de Desarrollo de Software. [En línea] 2014. [Citado el: 22 de 11 de 2013.] <http://es.scribd.com/doc/2050925/metodologias-de-desarrollo-software>.
- 15- BRESLAV. 2000. *A. Groovy and Grails*. 2000.
- 16-DREW. 2012. E. D. *METODOLOGÍA XP - EXTREME PROGRAMING (XP)*. 2012.
- 17- Laforge, Guillaume. 2014. Groovy. *Un lenguaje dinámico para la plataforma Java*. [En línea] 28 de 1 de 2014. [Citado el: 25 de 11 de 2013.] <http://groovy.codehaus.org/>.
- 18-Villalobos, Jorge. 2011. Código de Programación. *¿Qué es Grails? Framework de desarrollo web*. [En línea] 30 de 6 de 2011. [Citado el: 20 de 11 de 2013.] <http://codigoprogramacion.com/articulos/programacionweb/ique-es-grails-framework-de-desarrollo-web.html>.
- 19- Valle, Antonio. 2006. Gobierno TIC. Conocimiento Adquirido. *¿ServiceDesk o Gestión de Incidencias? Una queja no es una incidencia*. [En línea] 10 de 7 de 2006. [Citado el: 21 de 11 de 2013.] <http://www.gobiernotic.es/2006/07/servicedesk-o-gestin-de-incidencias.html>.
- 20-Avantare Consultores. 2014. Avantare. [En línea] 2014. [Citado el: 8 de 11 de 2013.] <http://www.avantare.com/lineas-de-negocio/materiales-de-referencia/que-es-cmmi-para-servicios-cmmi-svc?es>.
- 21-Layme Velázquez, Rubén Darío. 2011. *ISO 20000-Gestión de Servicios*. 2011.
- Osiatis. 2010. Itil V3. [En línea] 2010. http://itilv3.osiatis.es/ciclo_vida_servicios_TI.php.
- 22-Deutsch, K. W. (1988). *The Analysis of International Relations*. New York: Prentice-Hall Inc.- Division of Simon and Schuster.
- 23- Dybå Tore, Moe Nils Brede. (2004). *Process Improvement in practice, a Handbook for IT Companies*.
- 24-Harrington. (1994). *Mejoramiento de los procesos de la empresa*. McGraw Hill.
- 25-Nye, J. S. (1968). *Integración Regional Comparada: Concepto y Medición*.

- 26- Dybå Tore, Moe Nils Brede. (2004). *Process Improvement in practice, a Handbook for IT Companies*.
- 27-Harrington. (1994). *Mejoramiento de los procesos de la empresa*. McGraw Hill.
- 28-Nye, J. S. (1968). *Integración Regional Comparada: Concepto y Medición*.
- 29-OSIATIS. (2010). *econocom osiatis*. Obtenido de [http://itil.osiatis.es/Curso ITIL/Gestion Servicios TI/fundamentos de la gestion TI/que es ITIL/s_oporte al servicio.php](http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/fundamentos_de_la_gestion_TI/que_es_ITIL/s_oporte_al_servicio.php).
- 29- Romero, H. (2009). *Metodologías de Desarrollo*.
- .

Bibliografía

- 1-Harold Koontz, Heinz Weirich. 2004. *Administración Un Perspectiva Global*. s.l.: McGraw-Hill Interamericana, 2004.
- 2-Avantare Consultores. 2014. Avantare. [En línea] 2014. [Citado el: 8 de 11 de 2013.] <http://www.avantare.com/lineas-de-negocio/materiales-de-referencia/que-es-cmmi-para-servicios-cmmi-svc?es>.
- 3-BRESLAV. 2000. *A. Groovy and Grails*. 2000.
- 4-CA technologies. 2012. *CA Service Desk Manager*. 2012.
- 5-Coello, Helkyn. 2008. Helkyn Coello Blog. *ITIL, COBIT, CMMI, PMBOK: Como integrar y adoptar los estándares para un buen Gobierno de TI*. [En línea] 8 de 12 de 2008. [Citado el: 11 de 11 de 2013.] <http://helkyncoello.wordpress.com/2008/12/08/itil-cobit-cmmi-pmbok-como-integrar-y-adoptar-los-estandares-para-un-buen-gobierno-de-ti/>.
- 6-Corona, Mauricio. 2011. *Best Practices Gurus*. 2011.
- 7-Customer Care Associates. 2011. [En línea] 2011. http://www.customercareassoc.com/index.php?option=com_content&view=article&id=190%3Aoutsourcing-service-desk-una-opcion-para-alinearnos-al-negocio&catid=21%3Ageneral&Itemid=1.
- 8-DREW. 2012. *E. D. METODOLOGÍA XP - EXTREME PROGRAMING (XP)*. 2012.
- 9-Ireo. 2002. Ireo. [En línea] 2002. [Citado el: 13 de 11 de 2013.] [URL:http://www.ireo.com/fabricantes-y-productos/manageengine/servicedesk-plus/resumen/](http://www.ireo.com/fabricantes-y-productos/manageengine/servicedesk-plus/resumen/).
- 10-Laforge, Guillaume. 2014. Groovy. *Un lenguaje dinámico para la plataforma Java*. [En línea] 28 de 1 de 2014. [Citado el: 25 de 11 de 2013.] <http://groovy.codehaus.org/>.
- 11-Layme Velázquez, Rubén Darío. 2011. *ISO 20000-Gestión de Servicios*. 2011.
- 12- Osiatis. 2010. Itil V3. [En línea] 2010. http://itilv3.osiatis.es/ciclo_vida_servicios_TI.php.
- 13- Pía Ramírez Bravo, Felipe Donoso Jaurés. 2006. *METODOLOGÍA ITIL. Descripción, Funcionamiento y Aplicaciones*. Santiago : s.n., 2006.
- 14- Ramírez Bravo, Pía y DONOSO JAURÉS, FELIPE. 2006. *METODOLOGÍA ITIL*. Santiago : s.n., 2006.

- 15- Riverón, Yaíma Oval. 2007. *Servicio de Soporte Técnico utilizando la tecnología Service – Desk* . s.l. : UCI, 2007.
- 16- Silva, Reinaldo Oliveira Da. 2002. *Teorías de la Administración*. s.l. : International Thomson Editores, S.A. de C.V., 2002.
- 17- Stephen Robbins, Mary Coulter. 2005. *Administración*. s.l. : Pearson Educación, 2005.
- 18- Sun Microsystems. 2003. ITIL® IT Infrastructure Library. *Las Mejores Prácticas para la Gestión de Servicios de TI en su Organización*. [En línea] 2003. [Citado el: 10 de 11 de 2013.] sun.es/services/itil.
- 19- Totem. [En línea] <http://www.totemguard.com/netsupport-servicedesk/gestion-incidencias-helpdesk-preguntas-frecuentes/#P2>.
- 20- TotemGuard Digital Security. 2012. TotemGuard. Reduce en un 90 % el volumen de llamadas que recibe tu help desk. [En línea] 2012. <http://www.totemguard.com/netsupport-servicedesk/gestion-incidencias-helpdesk-preguntas-frecuentes/#P2>.
- 21- Valle, Antonio. 2006. Gobierno TIC. Conocimiento Adquirido. *¿ServiceDesk o Gestión de Incidencias? Una queja no es una incidencia*. [En línea] 10 de 7 de 2006. [Citado el: 21 de 11 de 2013.] <http://www.gobiernotic.es/2006/07/servicedesk-o-gestin-de-incidencias.html>.
- 22- Vilches, Ernesto. 2010. *Guía de Gestión de Servicios basada en Fundamentos de ITIL v3 © 2010*,. Madrid : Luarna Ediciones, 2010.
- 23- Villalobos, Jorge. 2011. Código de Programación. *¿Qué es Grails? Framework de desarrollo web*. [En línea] 30 de 6 de 2011. [Citado el: 20 de 11 de 2013.]
- 24- Deutsch, K. W. (1988). *The Analysis of International Relations*. New York: Prentice-Hall Inc.- Division of Simon and Schuster. .
- 25- Dybå Tore, Moe Nils Brede. (2004). *Process Improvement in practice, a Handbook for IT Companies*.
- 26- Harrington. (1994). *Mejoramiento de los procesos de la empresa*. McGraw Hill.
- 27- Nye, J. S. (1968). *Integración Regional Comparada: Concepto y Medición*.
- 28- OSIATIS. (2010). *Osiatis*. Obtenido de <http://itilv3.osiatis.es/itil.php>
- 29- Romero, H. (2009). *Metodologías de Desarrollo*.

Glosario de Términos

CMI: Cuadro de Mando Integral.

CMIS: Sistema de Información de Gestión de la Capacidad.

CSS: Hojas de Estilo en Cascada (CSS, siglas en inglés).

HTML: (lenguaje de marcado hipertextual), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTTP: Protocolo de transferencia de hipertexto.

JavaServer Pages: Es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML entre otros tipos de documentos.

JavaServlet: Los servlets son objetos que corren dentro y fuera del contexto de un contenedor de servlets (ejemplo: Tomcat) y extienden su funcionalidad.

OLA: Acuerdos de Nivel de Operación.

RFC: Petición de Cambio.

SLA: Acuerdos de Nivel de Servicio.

UC: Contrato de Soporte.

XML: (Lenguaje de Marcas Extensible) es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C).

Anexos

Anexo 1: Historias de usuario

Historia de Usuario	
Código: RF 2.1.	Nombre Historia de Usuario: Crear usuarios.
Modificación de Historia de Usuario Número: Ninguna.	
Referencia: Ninguna.	
Programador: Macdemis Liliana Pozo Guevara.	Iteración Asignada: 2.
Prioridad: Alta.	Puntos Estimados: 0.3 semanas.
Riesgo en Desarrollo: Alto.	Puntos Reales: 0.3 semanas.
Descripción: Crear un nuevo usuario local del sistema o uno haciendo uso de los datos de la base de datos LDAP.	
Observaciones: Mediante la integración del sistema a la base de datos LDAP de la universidad se obtienen los datos personales del nuevo usuario a crear.	
Prototipo de interfaz:	

Tabla 18. Historia de Usuario 2.1

Historia de Usuario	
Código: RF 2.2.	Nombre Historia de Usuario: Eliminar usuarios.
Modificación de Historia de Usuario Número: Ninguna.	
Referencia: Ninguna.	
Programador: Macdemis Liliana Pozo Guevara.	Iteración Asignada: 2.
Prioridad: Media.	Puntos Estimados: 0.1 semanas.
Riesgo en Desarrollo: Media.	Puntos Reales: 0.1 semanas.
Descripción: Eliminar un usuario de la base de datos.	
Observaciones:	
Prototipo de interfaz:	
<p>The screenshot shows a user management interface. At the top, there are three buttons: a green button with a plus sign and the text '+ Nuevo Usuario', a grey button with a person icon and the text 'Crear un Nuevo Usuario del Ldap', and a grey button with a list icon and the text 'Lista de Usuarios'. Below these buttons is a large heading 'Mostrar usuario'. Underneath the heading, there are three lines of user information: 'Usuario: 2', 'Habilitado: Si', and 'Roles: Técnico'. At the bottom of this section, there are two buttons: a blue button labeled 'Editar' and a red button labeled 'Eliminar'.</p>	

Tabla 19. Historia de Usuario 2.2

Anexo 2: Tarjetas CRC

Tarjeta CRC	
Clase: Aviso	
Responsabilidades	Colaboraciones
<p>Es la encargada de permitirle al usuario autenticado, publicar un aviso en la interfaz general del sistema.</p> <ul style="list-style-type: none"> ➤ String titulo ➤ String contenido ➤ Date desde_fecha ➤ Date fecha_fin ➤ String para ➤ String copia 	

Tabla 20. Tarjeta CRC Aviso

Tarjeta CRC	
Clase: Permiso	
Responsabilidades	Colaboraciones
<p>Es la encargada de definir los diferentes permisos para asignar los roles del sistema</p> <ul style="list-style-type: none"> ➤ String nombre ➤ static belongsTo = [modulo: Modulo] <pre>String toString() { nombre }</pre>	Modulo

Tabla 21. Tarjeta CRC Permiso

Tarjeta CRC	
Clase: Rol	
Responsabilidades	Colaboraciones
<p>Es la encargada de almacenar el nombre del rol y los permisos con que va a constar el mismo.</p> <ul style="list-style-type: none"> ➤ String nombre ➤ static hasMany = [permisos: Permiso] <pre>static mapping = { permisos lazy: false, joinTable: [name:'rol_permiso',key: 'idrol', column: 'idpermiso'] }</pre>	<p>Permiso</p>

Tabla 22. Tarjeta CRC Rol

Anexo 3: Pruebas de Aceptación

Caso de prueba de aceptación	
Código: HU 1_P1	Historia de usuario: 2.2
Nombre: Modificar usuario.	
Descripción: Modificar los atributos de un usuario almacenados en la base de datos del sistema.	
Condiciones de ejecución: El usuario logueado debe desempeñar el rol de administrador.	
Entradas/Pasos de ejecución: El administrador hace clic sobre el usuario a modificar El sistema muestra vista con los datos del usuario a modificar en la base de datos del sistema. Si se modifican de forma correcta los siguientes datos: <ul style="list-style-type: none"> • Nombre. • Activo. • Rol. El sistema muestra una vista con los datos del usuario ya modificados.	
Resultado de la prueba: Se modifica correctamente los datos del usuario seleccionado.	
Evaluación de la prueba: Bien.	

Tabla 23. Prueba de Aceptación de la Historia de Usuario 2.2

Caso de prueba de aceptación	
Código: HU 1_P1	Historia de usuario: 2.3
Nombre: Eliminar usuario.	
Descripción: Eliminar un usuario de la base de datos.	
Condiciones de ejecución: El usuario logueado debe desempeñar el rol de administrador.	
Entradas/Pasos de ejecución: El administrador hace clic sobre el usuario a eliminar El sistema muestra vista con los datos del usuario a eliminar en la base de datos del sistema. El usuario selecciona la funcionalidad de eliminar. El sistema muestra un mensaje para la confirmación El usuario confirma su solicitud. El sistema muestra una la vista con el listado de los usuarios y una alerta de que el usuario sido eliminado	
Resultado de la prueba: Se eliminado correctamente los datos del usuario seleccionado.	
Evaluación de la prueba: Bien.	

Tabla 24. Prueba de Aceptación de la Historia de Usuario 2.3

Caso de prueba de aceptación	
Código: HU 1_P1	Historia de usuario: 2.4
Nombre: Buscar usuario.	
Descripción: Buscar un usuario de la base de datos.	
Condiciones de ejecución: El usuario logueado debe desempeñar el rol de administrador.	
Entradas/Pasos de ejecución: El administrador introduce el nombre del usuario a buscar en la barra de navegación de la vista con la lista de los usuarios registrados El sistema muestra vista con él usuario que cumplió con el criterio de busca.	
Resultado de la prueba: Se encontró el usuario en la base de datos del sistema.	
Evaluación de la prueba: Bien.	

Tabla 25. Prueba de Aceptación de la Historia de Usuario 2.4

Anexo 4: Pruebas Unitarias



The screenshot displays the 'Unit Test Results - All tests' page. At the top left is a green icon of a person with arms raised. The main heading is 'Unit Test Results - All tests' with a sub-message: 'A single test executed without a single error or failure!'. On the right side, there are three links: 'Tests with failure and errors', 'Package summary', and 'Show all tests'. Below this, a green bar highlights the package 'TareaControllerTests' with the message 'A single test executed without a single error or failure!'. Underneath, a single test 'testCreate' is listed with a green checkmark icon and the text 'Executed in 21.673 seconds.'

Figura 16. Prueba Unitaria realizada al método Create de la clase TareaController



The screenshot displays the 'Unit Test Results - All tests' page. At the top left is a green icon of a person with arms raised. The main heading is 'Unit Test Results - All tests' with a sub-message: 'A single test executed without a single error or failure!'. On the right side, there are three links: 'Tests with failure and errors', 'Package summary', and 'Show all tests'. Below this, a green bar highlights the package 'TareaControllerTests' with the message 'A single test executed without a single error or failure!'. Underneath, a single test 'testsave' is listed with a green checkmark icon and the text 'Executed in 41.974 seconds.'

Figura 17. Prueba Unitaria realizada al método save de la clase TareaController

Anexo 5: Vistas del Sistema



Figura 18. Vista de la interfaz del Módulo de Administración

Centro de Soporte

Panel de Control Solicitudes Problemas Cambios Soluciones Contratos Bienvenido admin

Admin. Soporte

MÁS

- Comentarios
- Acera de
- Ayuda

Tablero Catálogo de servicio Programa

Mi Resumen

Solicitudes vencidas	0
Solicitudes programadas para hoy	0
Solicitudes abiertas	0
Solicitudes aprobadas	0
Cambios no aprobadas	0
Problemas abiertos	0
Problemas no asignados	0

Mis Tareas

prueba Estado: Abierta

+ Agregar nueva Mostrar Todo

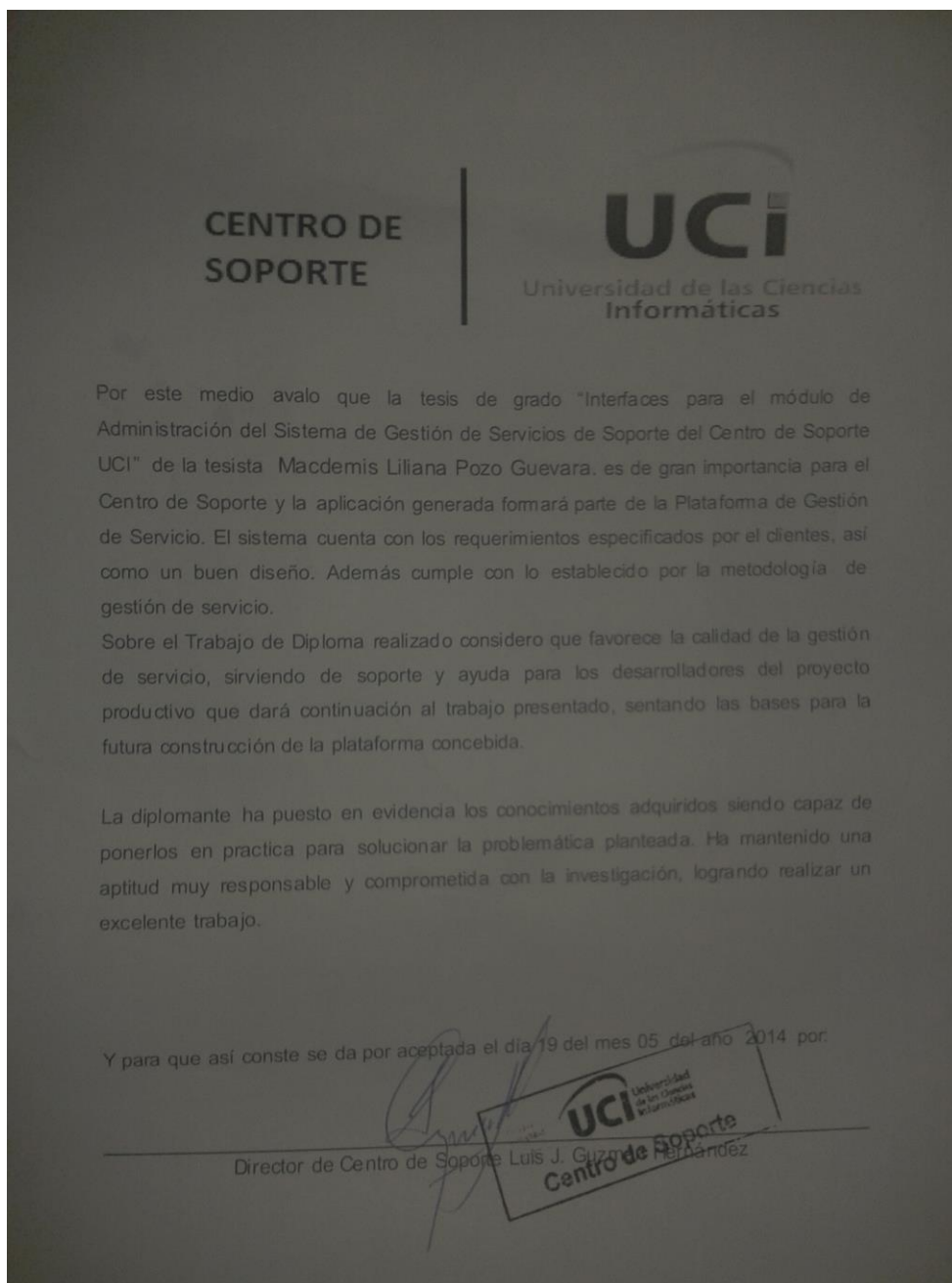
Avisos

+ Agregar nueva Mostrar Todo

presentarse - estar todos mañana 3 pm en el lab

Universidad de las Ciencias Informáticas

Figura 19. Vista de la interfaz General del Sistema de Gestión de Servicios de Soporte

Anexo 6: Aval del centro de soporte UCI

Aval del Centro de Soporte UCI a las Interfaces del Módulo de Administración para el Sistema de Gestión de Servicios