

Universidad de las Ciencias Informáticas
Facultad 1



**TÍTULO: SUBSISTEMA DE GESTIÓN DE CATÁLOGOS Y GESTIÓN DE NIVEL
DE SERVICIO DEL CENTRO DE SOPORTE UCI.**

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



***Autores: Maidelis Vega Calzado.
Emil Pérez Piz.***

***Tutores: MSc Yulio Seriocha García.
Ing. Neybis Lago Clara.***

“La Habana, Junio del 2014”



"Hay en el mundo un lenguaje que todos comprenden. es el lenguaje del entusiasmo, de las cosas hechas con amor y con voluntad, en busca de aquello que se desea o en lo que se cree."

Paulo Coelho

Declaración de Autoría

Declaramos que somos los únicos autores del presente trabajo que tiene como título Subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio del Centro de Soporte UCI y autorizamos al Centro de Soporte de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmamos la presente a los __10__ días del mes de __junio__ del año __2014__.

Maidelis Vega Calzado

(Autora)

Emil Pérez Piz

(Autor)

MSc Yulio Seriocha García

(Tutor)

Ing. Neybis Lago Clara

(Tutora)

Agradecimientos

Gracias mami, por ser mi motor impulsor, mi mayor orgullo, por no rendirte nunca, por ser mi ejemplo de superación y mi inspiración para querer ser mejor cada día, por tus sabias palabras en el momento adecuado, por tu cariño infinito, tu confianza, porque sin ti este logro no fuera posible, te amo mami.

Gracias papi, por cuidar a mami, por todos tus años de sacrificio para darme lo necesario en aras de alcanzar este sueño, por ser mi ejemplo de perseverancia, por saber levantarte ante situaciones difíciles, por cada consejo, por hacer tuyos mis problemas, porque a pesar de no tener tu sangre me has acogido como tu hija y has tolerado todas mis chiquilladas, siempre seré tu niña malcriada, te quiero un montón.

Gracias mi tata, por quererme tanto, por ser tu ejemplo a seguir, porque a pesar de tener una hermana insoportable (como dice mami) has sabido entenderme y darme una sonrisa cuando más lo necesito, disculpa si no demuestro todo el amor que llevo dentro, disculpa cada pelea, eres el hermanito que más quiero.

A mi tía Miladys y Eliecer, por ser mis segundos padres, por acogerme en su casa y darme todo el apoyo necesario, gracias por cuidar de mí. A toda mi familia, especialmente a mis hermanitas y a mis abuelas, por ser parte de este sueño y darle un poco de amor a mi vida.

A mi tutora Neybis por todo el apoyo brindado, por su preocupación, por dedicarme tiempo y tener mucha paciencia. A la profesora Elieyis, gracias por ayudarme siempre que lo necesitaba, por siempre estar dispuesta a revisar el documento, por nunca decir que "no". A mi compañero de tesis, por todo su apoyo, por ser tan responsable y alentarme en los momentos de desesperación, por soportarme, por hacerme reír con sus ocurrencias y su música. Gracias por confiar en mí, Phoco.

A Roxey por ser mi sister, por cada consejo, por todos sus regaños y peleas, por cada risa, cada fiesta compartida, por su gran paciencia para soportar mis malcriadeces y mis cambios de luna, te quiero mucho aunque lo demuestre poco. A mis amigos May y Gatico por llenarme de buenos momentos, por sus consejos en los momentos precisos, por hacerme reír con sus ocurrencias, los quiero de aquí a Plutón. A mis compañeros de grupo, especialmente Dayán, Pedri, Estrada, Benito, El indio, gracias por todos los momentos de alegría. A mis viejas y buenas amistades, Lily, Sune, Paili, a pesar de estar un poco separadas las quiero un montonazo, gracias por su preocupación.

A todas las personas que se preocuparon por mí, durante el desarrollo de este trabajo y a todas las que han influenciado en mi formación durante estos 5 años. Gracias...

Maidelis Vega Palzado

Agradecimientos

A mi madre, el tesoro más grande de mi castillo, gracias por creer en mí y apoyarme en todo momento, quiero que sepas que este momento solo es un insignificante regalo que te entrego comparado con todos lo que te daré en la vida.

A mi padre, mi faro y ejemplo a seguir, gracias por tu apoyo, porque sé que aunque no me lo hagas saber siempre estás pendiente de todo lo que me sucede, ahora después de este día, seré yo, el que estará pendiente de usted y le prometo que nada le faltará.

A mi flaquita hermosa, por aconsejarme y apoyarme en todo momento, incluso cuando he dejado de tener la razón, gracias por todo el cariño que me entregas, te amo.

A mi familia, a mis amigos, a todas las personas que en algún momento me preguntaron ¿cómo va esa tesis?, a todos los que me brindaron su apoyo y ayuda desinteresada, a mi compañera de trabajo, por ser tan dedicada y laboriosa y por último y no por menos importante, a los tutores, gracias.

Emil Pérez Piz

Dedicatoria

Quisiera dedicar el presente trabajo a todas las personas especiales e importantes para mí, las que me han apoyado y cuidado durante estos 5 años y a las cuales no imagino lejos. A mami por ser la personita que más admiro en el mundo, por ser tan comprensiva y paciente conmigo. A papi, por darme tantas fuerzas, guiarme y enseñarme a ser una mejor persona. A mis hermanitos Redrito, Sandrita y Leydis por ser partecitas de mi vida, espero poder ser un ejemplo para ellos. A Filian y Daneis por creer en mí, por estar siempre conmigo y por demostrarme que aún existen los amigos verdaderos. A Faily, porque a pesar de estar lejos siempre pude contar con ella en todo momento. En fin, a toda mi familia y amigos por su apoyo incondicional durante la realización de este trabajo, especialmente a mis abuelitas Martha y Clara.

De Maidelis

Dedico esta Tesis a mis queridos padres por ser un ejemplo de rectitud amor y confianza, gracias a ellos he logrado alcanzar una parte considerable de las metas que me he propuesto. A mi novia, por permanecer a mi lado y llegar a convertirse no solo en mi compañera, sino en mi amiga y confidente.

De Emil

Resumen

El Centro de Soporte de la Universidad de las Ciencias Informáticas, es el encargado de prestar servicios de soporte técnico a todos los productos informáticos desarrollados en los centros productivos de la Universidad. Para ello cuenta con una plataforma propietaria, la cual impide la actualización de los servicios según las necesidades existentes en el centro. Además, los involucrados no pueden consultar los Acuerdos de Nivel de Servicio, los Acuerdos de Nivel de Operación y los Contratos de Soporte, afectando la calidad de los servicios porque se desconocen las necesidades de los clientes. Por esta razón se desarrolló un subsistema de Gestión de Catálogos y Gestión de Nivel de Servicios para contribuir al correcto funcionamiento de estos procesos en el Centro de Soporte. Para el desarrollo del subsistema se utilizó Grails como marco de trabajo sobre el lenguaje de programación Groovy, IntelliJ IDEA como IDE de programación, Apache Tomcat como servidor web, PostgreSQL como gestor de bases de datos y Visual Paradigm como herramienta para el modelado. El proceso de desarrollo de *software* fue guiado por la metodología ágil Programación Extrema, generando los artefactos que la misma propone. Se realizaron las pruebas unitarias, de funcionalidad, de rendimiento y de integración para garantizar la calidad de la solución propuesta y reducir el número de errores. Este subsistema permite la creación de catálogos, la selección del servicio que se ajuste a las necesidades del cliente y garantiza los niveles de calidad de los servicios de Tecnología de Información prestados.

Palabras claves: Acuerdos de Nivel de Operación, Acuerdos de Nivel de Servicio, Tecnologías de la Información, Gestión de Catálogos, Gestión de Niveles de Servicios.

INTRODUCCIÓN	1
FUNDAMENTACIÓN TEÓRICA DEL SUBSISTEMA DE GESTIÓN DE CATÁLOGOS Y GESTIÓN DE NIVEL DE SERVICIO.	6
1.1. <i>Conceptos fundamentales</i>	6
1.2. <i>Estándares basados en marcos de referencia para la Gestión de Servicios de la Tecnología de la Información</i>	7
1.2.1. <i>Objetivos de Control para la Información y Tecnologías</i>	8
1.2.2. <i>Organización Internacional de Normalización/Comisión Electrotécnica Internacional 20000</i>	9
1.2.3. <i>Integración de Modelos de Madurez de Capacidades para Servicios</i>	10
1.2.4. <i>Biblioteca de Infraestructura de Tecnologías de Información</i>	10
1.2.5. <i>Análisis crítico del estándar a utilizar</i>	11
1.3. <i>Fases del ciclo de vida de la Biblioteca de Infraestructura de Tecnologías de Información</i>	12
1.4. <i>Procesos de la Biblioteca de Infraestructura de Tecnologías de Información</i>	13
1.4.1. <i>Proceso Gestión de Niveles de Servicio según la Biblioteca de Infraestructura de la Tecnología de Información</i>	13
1.4.2. <i>Proceso Gestión de Catálogos de Servicio según la Biblioteca de Infraestructura de la Tecnología de Información</i>	15
1.5. <i>Ventajas de la Biblioteca de Infraestructura de Tecnologías de Información para la Tecnología de Información</i>	17
1.6. <i>Ventajas al adoptar la Biblioteca de Infraestructura de Tecnologías de Información para la Tecnología de Información</i>	17
1.6.1. <i>Integración de estándares</i>	18
1.7. <i>Análisis de soluciones existentes</i>	19
1.7.1. <i>Remedy IT Service Management</i>	19
1.7.2. <i>HP Service Manager</i>	20
1.7.3. <i>IBM Tivoli Service Request Manager</i>	20
1.7.4. <i>ManageEngine ServiceDesk Plus</i>	20
1.7.5. <i>Análisis crítico de las soluciones estudiadas</i>	20
1.8. <i>Metodología de desarrollo de software</i>	21
1.9. <i>Herramientas y tecnologías utilizadas</i>	23
1.9.1. <i>Groovy and Grails</i>	23
1.9.2. <i>IntelliJ IDEA</i>	23
1.9.3. <i>Apache Tomcat</i>	24
1.9.4. <i>PostgreSQL</i>	24
1.9.5. <i>BootStrap</i>	24
1.9.6. <i>Visual Paradigm</i>	25
1.9.7. <i>JMeter</i>	25
1.10. <i>Conclusiones parciales</i>	25
ANÁLISIS Y DISEÑO DEL SUBSISTEMA DE GESTIÓN DE CATÁLOGOS Y GESTIÓN DE NIVEL DE SERVICIO	26
2.1 <i>Solución propuesta</i>	26
2.2 <i>Actores del sistema</i>	27
2.3 <i>Modelo conceptual</i>	28
2.4 <i>Requisitos del sistema</i>	29
2.4.1 <i>Requisitos funcionales</i>	29
2.4.2 <i>Requisitos no funcionales</i>	33
2.5 <i>Fase de exploración</i>	35

2.5.1	Historias de Usuario	35
2.6	<i>Fase de planificación</i>	37
2.6.1	Estimación de esfuerzo	37
2.6.2	Plan de duración de iteraciones.....	39
2.6.3	Plan de entrega	41
2.7	<i>Diseño del sistema</i>	42
2.7.1	Tarjetas CRC del sistema	42
2.7.2	Patrones de diseño utilizados	43
2.7.3	Patrón de arquitectura	44
2.8	<i>Diseño de la base de datos</i>	45
2.9	<i>Conclusiones parciales</i>	46
IMPLEMENTACIÓN Y VALIDACIÓN DEL SUBSISTEMA DE GESTIÓN DE CATÁLOGOS Y GESTIÓN DE NIVEL DE SERVICIO		47
3.1	<i>Implementación del sistema</i>	47
3.1.1	<i>Estándares de codificación</i>	47
3.2	<i>Pruebas de software</i>	53
3.2.1	<i>Pruebas unitarias</i>	53
3.2.3	<i>Pruebas de rendimiento</i>	58
3.2.4	<i>Pruebas de integración</i>	59
3.3	<i>Conclusiones parciales</i>	60
Conclusiones		61
Recomendaciones		62
Bibliografía referenciada		63
Bibliografía consultada.....		66
Glosario de Términos.....		69
Anexo 1: Carta de aceptación del cliente		70
Anexo 2: Guía de observación		71
Anexo 3: Entrevista.....		72
Anexo 4: Lista de Reserva del Producto.		73
Anexo 5: Modelo de datos		78
Anexo 6: Prueba de rendimiento.....		79
Anexo 7: Vistas de la solución propuesta.....		80

<i>Figura 1: Ciclo de Vida de los Servicios de TI (OSIATIS S.A 2007a)</i>	12
<i>Figura 2: Fases y procesos de ITIL 2011 (Corona 2011)</i>	13
<i>Figura 3: Ciclo de la Gestión de Nivel de Servicio (OSIATIS S.A 2013)</i>	14
<i>Figura 4: Interacciones y funcionalidades del Catálogo de Servicios (OSIATIS S.A 2007b)</i>	16
<i>Figura 5: Modelo Conceptual</i>	28
<i>Figura 6: Estructura del patrón MVC</i>	45
<i>Figura 7: Uso de la notación CamelCase</i>	48
<i>Figura 8: Uso de la notación Snake-case</i>	48
<i>Figura 9: Uso de la notación Pascal Casing</i>	48
<i>Figura 10: Prueba unitaria aplicada a la clase RollController</i>	54
<i>Figura 11: Prueba unitaria aplicada a la clase Usuario</i>	54
<i>Figura 12: Prueba unitaria aplicada a la clase Categorías</i>	54
<i>Figura 13: Resultado de las pruebas funcionales</i>	58
<i>Figura 14: Resultados obtenidos para 300 peticiones concurrentes</i>	59
<i>Figura 15: Carta de aceptación del cliente</i>	70
<i>Figura 16: Modelo de datos de la solución propuesta</i>	78
<i>Figura 17: Prueba de rendimiento para 300 peticiones concurrentes</i>	79
<i>Figura 18: Vista principal de la solución propuesta</i>	80
<i>Figura 19: Vista administrativa de la solución propuesta</i>	80
<i>Figura 20: Gestión de cuentas de usuario de la solución propuesta</i>	81

Tabla 1: Actores del sistema.....	27
Tabla 2: HU Insertar Servicio.....	36
Tabla 3: HU Insertar área funcional.....	37
Tabla 4: Estimación de esfuerzo por HU.....	39
Tabla 5: Descripción de iteraciones.....	39
Tabla 6: Plan de duración de iteraciones.....	41
Tabla 7: Plan de Entrega.....	42
Tabla 8: Tarjeta CRC de ANSController.....	42
Tabla 9: Tarjeta CRC de ANS.....	43
Tabla 10: Tarea de ingeniería 1 para la HU Insertar servicio.....	50
Tabla 11: Tarea de ingeniería 2 para la HU Insertar servicio.....	50
Tabla 12: Tarea de ingeniería 3 para la HU Insertar servicio.....	50
Tabla 13: Tarea de ingeniería 1 para la HU Insertar evaluación.....	51
Tabla 14: Tarea de ingeniería 2 para la HU Insertar evaluación.....	51
Tabla 15: Tarea de ingeniería 3 para la HU Insertar evaluación.....	52
Tabla 16: Tarea de ingeniería 1 para la HU Asociar una evaluación.....	52
Tabla 17: Caso de prueba 1 para la HU Insertar servicio.....	56
Tabla 18: Caso de prueba 2 para la HU Insertar servicio.....	56
Tabla 19: Caso de prueba 1 para la HU Insertar área funcional.....	57
Tabla 20: Caso de prueba 2 para la HU Insertar área funcional.....	58
Tabla 21: Lista de reserva del producto.....	77

INTRODUCCIÓN

En la actualidad el uso de la informática se ha convertido en un elemento de suma importancia para el desarrollo de la sociedad, constituyendo una pieza clave en el éxito de una organización. Con el avance de esta ciencia ha aumentado la competitividad entre las organizaciones y con ella el incesante uso de las Tecnologías de Información (por sus siglas en inglés IT) para disminuir las ventajas de los competidores, o bien, para mantener e incrementar las ventajas competitivas. Es importante agregar que las IT se definen como *“todas las herramientas y métodos empleados para recabar, retener, manipular o distribuir información”* (Bologna and Wash 1997).

La necesidad de incrementar la cantidad y uso de las IT viene impuesta, entre otras causas, por la constante búsqueda de productividad, calidad y continua renovación. Además, puede ser utilizada para extenderse más allá de antiguos límites temporales, geográficos y de tamaño; posibilitando notables cambios en las organizaciones. *“Su uso inteligente y apropiado ha ocasionado una profunda transformación de las mismas e incluso de los mercados, de manera que la información se ha convertido en un activo cada vez más valorado, razón por la cual hoy las organizaciones hacen inversiones importantes en recursos de IT”* (Carrillo Verdún and Rubio Casallas 2011).

La práctica ha demostrado que las IT por sí solas no pueden satisfacer las necesidades reales de la empresa y los clientes. Es de suma importancia la utilización de otros conceptos importantes como la Gestión de Servicios de Tecnología de Información (ITSM por sus siglas en inglés), para aumentar la calidad de los servicios que proveen las IT y para lograr mejores relaciones con los clientes. Esto se debe a que *“es una disciplina basada en procesos, enfocada en alinear los servicios de IT proporcionados con las necesidades de las empresas, percibir el cliente final, proporcionando el aumento de la eficiencia y reduciendo los riesgos asociados a los servicios de IT”* (Calafat Mesquida 2012).

Sin embargo, han surgido diversas problemáticas al gestionar las IT, entre las principales, se encuentran la mala gestión de los proyectos, la gestión inadecuada de servicios, falta de procesos de control y monitorización, métodos de desarrollo de *software* inadecuados, falta de alineamiento estratégico para las iniciativas IT, entre otras (Coello 2008). Estas problemáticas pueden ser solucionadas mediante el uso de múltiples estándares basados en marcos de referencia para la ITSM, aunque no todos se ajustan con otros porque fueron creados en contextos y con propósitos diferentes (Carrillo Verdún and Rubio Casallas 2011). Por estas razones es primordial saber elegir los modelos adecuados, existen guías prácticas como la Organización Internacional de Normalización/Comisión Electrotécnica Internacional 20000 (por sus siglas en inglés ISO/IEC 20000), Modelos de Madurez de Capacidades para Servicios (por sus siglas en inglés CMMI SVC), marcos de gobiernos como Objetivos de Control para la Información y Tecnologías

(por sus siglas en inglés COBIT) y mejores prácticas como la Biblioteca de Infraestructura de Tecnologías de Información (por sus siglas en inglés ITIL) permitiendo la mejora de los servicios de IT.

COBIT es un modelo de evaluación y monitoreo que enfatiza en el control de negocios y la seguridad IT, abarca controles específicos de IT desde una perspectiva de negocio. Su primera edición fue publicada en 1996 por el instituto de Control de IT y la Asociación de Auditoría y Control de Sistemas de Información (ISACA por sus siglas en inglés) (Área de auditoría y control 2007). Su principal propósito es definir una estructura que comprenda, implante y evalúe capacidades, rendimiento y riesgos de IT para cumplir los requisitos del negocio.

ISO/IEC 20000 es el primer estándar específico para la ITSM. Su principal objetivo es aportar requisitos necesarios dentro del marco de un sistema completo e integrado, permitiéndole a una organización proveer servicios IT gestionados, de calidad y que satisfagan los requisitos de negocio de los clientes (Calafat Mesquida 2012). Esta norma es aplicable a cualquier organización que se base en servicios de IT, sin importar su tamaño o sector y proporciona la base para probar que una organización de IT ha implantado buenas prácticas para la gestión de servicio y que las está usando de forma regular y consistente.

CMMI SVC es un conjunto de mejores prácticas, áreas de proceso, para los proveedores de servicios, normalmente, de IT. También proporciona un marco para la mejora de los procesos relacionados con los servicios, clasificando sus áreas en 5 niveles de madurez (Jaramillo González, Trejo Ramírez and Barranco García 2010).

ITIL, por su parte, es el estándar más ampliamente conocido para la ITSM (Coello 2008). Propone el establecimiento de estándares para ayudar en el control, operación y administración de los recursos. Plantea hacer una revisión y reestructuración de todos los procesos existentes en caso de que estos los necesiten, lo que conduce a una mejora continua. Según (Carrillo Verdún and Rubio Casallas 2011) *“este conjunto de buenas prácticas 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 y tecnología”*.

ITIL 2011 ofrece una visión global de la vida de un servicio desde su diseño hasta su retirada de producción, sin ignorar todos los procesos y funciones que están involucrados en la eficiente prestación del mismo, estructura la ITSM sobre el concepto de Ciclo de Vida de los Servicios, complementándose en cinco fases: Estrategia del servicio, Diseño del servicio, Transición del servicio, Operación del servicio y Mejora continua del servicio respectivamente. Cada una de estas fases está compuesta por procesos indicados a distintas partes de la ITSM.

El proceso de Gestión de Catálogos tiene como objetivo crear y mantener el catálogo de servicios de la organización IT incluyendo toda la información relevante: gestores, estatus, proveedores, entre otras. La elaboración de este catálogo es imprescindible porque sirve de guía a los clientes a la hora de seleccionar un servicio que se adapte a sus necesidades, delimita las funciones y compromisos de la organización IT. Otro proceso importante es la Gestión de Niveles de Servicio, responsable de acordar y garantizar los niveles de calidad de los servicios IT prestados, poniendo la tecnología al servicio del cliente. Este proceso debe velar por la calidad de los servicios IT alineando tecnología con procesos de negocio y todo ello a un coste razonable.

Con el objetivo de informatizar el país y desarrollar la industria del *software* para contribuir al avance económico del mismo, se crea en el 2002 la Universidad de las Ciencias Informáticas (UCI), idea del Comandante en Jefe Fidel Castro Ruz. La UCI es una universidad productiva, cuya misión es producir *software* y servicios informáticos a partir de la vinculación estudio-trabajo e introduciendo en la práctica social las IT. La producción se concentra en el desarrollo de proyectos en diferentes centros productivos. Dentro de estos, se encuentra el Centro de Soporte UCI cuyo objetivo es prestar servicios de soporte técnico a las aplicaciones informáticas y productos desarrollados por cada centro productivo de la Universidad.

El Centro de Soporte, para su desempeño, es dependiente del buen funcionamiento de Atención al cliente, Soporte al Servicio, Provisión del Servicio, entre otras áreas dentro del flujo de trabajo que desarrolla. En la actualidad dicho flujo se ve interrumpido por inconvenientes en algunos de sus procesos vitales, estos son:

- La realización incompleta de la Gestión de Niveles de Servicios.
- Falta de actualización del Catálogo de Servicios.
- La plataforma que se utiliza para el soporte de los sistemas informáticos es propietaria.
- Los Acuerdos de Nivel de Servicio (por sus siglas en inglés SLAs)¹, los Acuerdos de Nivel de Operación (por sus siglas en inglés OLAs)² y los Contratos de Soporte (por sus siglas en inglés UCs)³ no se encuentran informatizados.

Estos inconvenientes traen como consecuencia:

- Imposibilidad de consultar los SLAs, OLAs y UCs a través de la plataforma.
- Incumplimiento en los tiempos de respuestas y resolución establecidos con el cliente.

¹ Recoge en un lenguaje no técnico todos los detalles de los servicios brindados.

² Documento interno de la organización que especifica las responsabilidades en la prestación de un servicio determinado.

³ Acuerdo con un proveedor externo para la prestación de servicios no cubiertos por la organización TI.

- Desconocimiento, por parte de los clientes, de las ofertas con las que cuenta el Centro de Soporte.
- Incorrecto almacenamiento de información vital para la contratación de servicios y su falta de disponibilidad.
- Mala selección, por parte del cliente, de los servicios que se adecuen a sus necesidades.
- Malentendido entre los implicados.

Por todo lo anteriormente expuesto y para dar solución a la necesidad existente, se plantea el siguiente **problema de investigación**: ¿Cómo contribuir al correcto funcionamiento de los procesos Gestión de Catálogos y Gestión de Nivel de Servicio?

Centrándose la investigación en el **objeto de estudio**: Los procesos de Gestión de Catálogos de Servicios y Gestión de Niveles de Servicio.

Para darle solución a la problemática planteada, se define como **objetivo general**: Desarrollar un subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio para contribuir al correcto funcionamiento de estos procesos en el Centro de Soporte UCI.

Enmarcado en el **campo de acción**: Los procesos de Gestión de Catálogos de Servicios y Gestión de Niveles de Servicio en el Centro de Soporte UCI.

Se define como **idea a defender**: El desarrollo de un subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio contribuirá al correcto funcionamiento de estos procesos en el Centro de Soporte UCI.

A partir del objetivo general definido, se proponen las siguientes **tareas de investigación**:

- Realización del diseño teórico de la investigación.
- Análisis del estado del arte sobre la Gestión de Catálogos de Servicios y Gestión de Niveles de Servicio.
- Análisis de las herramientas y tecnologías a utilizar en el proceso de desarrollo.
- Realización del análisis y diseño de la solución.
- Realización del modelo de implementación.
- Implementación de los componentes del modelo.
- Diseño y ejecución de los casos de prueba de las funcionalidades implementadas.

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

Métodos Teóricos:

Análisis histórico-lógico: Este método se utilizó en el análisis de documentos, teorías y otros materiales sobre la Gestión de Catálogos y Gestión de Niveles de Servicio, de manera que se procese la información y se elaboren conclusiones relacionadas con el objeto de estudio.

Analítico-sintético: Este método se utilizó para analizar y comprender la teoría y documentación relacionada con Gestión de Catálogos de Servicios y Gestión de Niveles de Servicio, permitiendo así, extraer los elementos más relacionados e importantes con el objeto de estudio.

Métodos Empíricos:

Observación: Este método permitió percibir a partir de la situación real que se está investigando cómo se desarrollan los procesos de Gestión de Catálogos de Servicios y Gestión de Niveles de Servicio. Se utilizó para realizar el diagnóstico acerca de la realización del sistema propuesto.

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

El presente trabajo de diploma está estructurado de la siguiente manera:

Consta de una introducción donde se ve presente el diseño de la investigación y actualidad del tema. Tres capítulos, conclusiones, recomendaciones y bibliografía utilizada. Además de un glosario de términos, donde se explican en detalles los términos técnicos que han sido utilizados en la elaboración del documento. Finalmente, los Anexos que complementan el trabajo realizado.

En el **Capítulo 1 Fundamentación teórica del Subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio del Centro de Soporte UCI**, se definen elementos teóricos que sustentan la investigación, se hace referencia a los conceptos fundamentales que permiten entender cualquier concepto asociado a la comprensión del trabajo. Se realiza un estudio del estado del arte de soluciones, dirigidas a resolver el problema de investigación planteado y se definen las herramientas y tecnologías a utilizar.

En el **Capítulo 2 Análisis y diseño del Subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio del Centro de Soporte UCI**, se describen las principales características y funcionalidades de la solución propuesta, se especifican los requisitos funcionales y no funcionales según necesidades del cliente y se presenta el diseño del subsistema.

Finalmente, en el **Capítulo 3 Implementación y validación del Subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio del Centro de Soporte UCI**, se identifican las tareas de ingeniería en las que se descompone una Historia de Usuario, se le realizan pruebas a la solución propuesta y luego se procede a validarla para verificar que cumple con el objetivo planteado, concluyendo con el resultado de dicha validación.

FUNDAMENTACIÓN TEÓRICA DEL SUBSISTEMA DE GESTIÓN DE CATÁLOGOS Y GESTIÓN DE NIVEL DE SERVICIO.



Actualmente, las organizaciones dependen cada vez más de los avances tecnológicos en el campo de la informática para lograr sus objetivos de negocio. Estos avances han producido una verdadera revolución en la producción, la prestación de servicios, la forma que se organizan y dirigen los procesos en una organización. Por esta razón, es de suma importancia el dominio de las nuevas tecnologías para obtener ventajas competitivas y para garantizar la máxima satisfacción del cliente. En la UCI, específicamente en el Centro de Soporte se llevan a cabo diferentes procesos enfocados en conocer las necesidades del cliente, para brindarles servicios de soporte técnico que satisfagan sus necesidades, siendo la gestión de catálogos y la gestión de nivel de servicio elementos claves.

El presente capítulo tiene como objetivo tratar los conceptos y aspectos significativos relacionados con catálogo y nivel de servicio, abordados en diferentes fuentes bibliográficas. Además, se realiza un estudio de algunos sistemas de Gestión de Catálogos de Servicios y Gestión de Niveles de Servicio existentes en Cuba y el mundo. Se proporciona una panorámica general acerca de los estándares basados en marcos de referencia para la ITSM, especialmente ITIL incluyendo un profundo análisis de su integración con otros estándares.

1.1. Conceptos fundamentales

Para lograr un mayor entendimiento de la problemática a resolver, es necesario fundamentar algunos conceptos básicos de suma importancia para el desarrollo del tema tratado en este trabajo. Por un lado la argumentación del término catálogo y por otra la del término nivel de servicio.

Catálogo

Según (Real Academia Española 2001) el término catálogo se define como “una relación ordenada en la que se incluyen o describen de forma individual libros, documentos, personas, objetos, etc., que están relacionados entre sí”.

Otras fuentes, como el diccionario de definiciones ABC definen a catálogo como una *“lista ordenada o clasificada que se hará sobre cualquier tipo de objetos (monedas, bienes a la venta, documentos, entre otros) o en su defecto personas y también catálogo será aquel conjunto de publicaciones u objetos que se encuentran clasificados normalmente para la venta”* (Definición ABC 2007).

La norma UNE 50-113-91-/2 define catálogo como *“un documento secundario que registra y describe documentos reunidos de forma permanente o temporal”* (VV.AA 2006).

(Hanna and Rance 2011) definen este término como *“una base de datos o un documento estructurado con información sobre todos los servicios de IT en producción, incluye los servicios que están disponibles para su implementación”*.

Después de realizar un análisis de las definiciones anteriores los autores asumen la definición de (Hanna and Rance 2011) por considerar que es la más ajustada al objeto de estudio.

Nivel de Servicio

En una de sus acepciones (Real Academia Española 2001) define el término nivel como *“medida de una cantidad con referencia a una escala determinada”*.

El término servicio (Definición ABC 2007) lo define como *“un conjunto de actividades que lleva a cabo internamente una empresa para poder responder y satisfacer las necesidades de un cliente”*.

Otras fuentes como (Real Academia Española 2001) definen este término como *“prestación humana que satisface alguna necesidad social y que no consiste en la producción de bienes materiales”*.

(Hanna and Rance 2011) plantean que nivel de servicio es *“el logro medido y notificado en informes con respecto a uno o más objetivos de nivel de servicio”*.

Después de realizar un análisis de las definiciones anteriores los autores asumen la definición de (Hanna and Rance 2011) por considerar que es la más ajustada al objeto de estudio.

1.2. Estándares basados en marcos de referencia para la Gestión de Servicios de la Tecnología de la Información.

Actualmente la industria de IT requiere mejorar la administración de la calidad y la confiabilidad de IT en los negocios, además de responder a un creciente número de requerimientos regulatorios y contractuales, debido a esto se hace imprescindible el uso de estándares y mejores prácticas tales como COBIT, ISO/IEC 20000, CMMI SVC e ITIL (IT Governance Institute 2008a), permitiendo una mejor gestión de IT, un gobierno eficaz de las actividades de IT, un marco de referencia eficaz para la gestión de políticas, controles internos y prácticas definidas, muchos otros beneficios, incluyendo ganancia de eficiencias, menor dependencia de expertos, menos errores, mejora de la confianza de los socios de negocios y de reguladores. A continuación se enuncian las principales características de estos estándares basados en marcos de referencia para la ITSM.

1.2.1. Objetivos de Control para la Información y Tecnologías

COBIT es un estándar generalmente aceptado que brinda buenas prácticas para la gestión y control de las IT, basado en estándares de la industria y las mejores prácticas. Una vez implementado, los ejecutivos pueden asegurar que se ajusta de manera eficaz con los objetivos del negocio y dirigir mejor el uso de IT para obtener ventajas comerciales (IT Governance Institute 2008b). Brinda las mejores prácticas y herramientas para el monitoreo y la gestión de las actividades de IT. Este estándar se enfoca más hacia la auditoría del cumplimiento de los procesos IT con los estándares de autoridades, el control sobre las funciones de IT, la medición y la gestión del riesgo. El marco de trabajo COBIT se creó con las características principales de ser orientado a negocios, orientado a procesos, basado en controles e impulsado por mediciones.

Este marco de trabajo resulta de interés a distintos usuarios, los cuales son enunciados a continuación (Sallé 2004):

- Dirección ejecutiva: Para obtener valor de las inversiones, riesgos de IT y para controlar la inversión en un ambiente de IT con frecuencia impredecible.
- Gerencia del negocio: Para obtener certidumbre sobre la administración y control de los servicios de IT, proporcionados internamente o por terceros.
- Gerencia de IT: Para proporcionar los servicios de IT que el negocio requiere para dar soporte a la estrategia del negocio de una forma controlada y administrada.
- Auditores: Para respaldar sus opiniones y/o para proporcionar asesoría a la gerencia sobre controles internos.

Los beneficios de implementar COBIT como marco de referencia de gobierno sobre la IT incluyen (IT Governance Institute 2006):

- Mejor alineación, con base en su enfoque de negocios.
- Una visión entendible para la gerencia, de lo que hace IT.
- Propiedad y responsabilidades claras, con base en su orientación a procesos.
- Aceptación general de terceros y reguladores.
- Entendimiento compartido entre todos los participantes en un lenguaje común.

1.2.2. Organización Internacional de Normalización/Comisión Electrotécnica Internacional 20000

ISO/IEC 20000 es un estándar de calidad de procesos de gestión de servicios de IT. Promueve la adopción de un enfoque de procesos integrados, para una provisión eficaz de servicios gestionados que satisfaga los requisitos del negocio y de los clientes (Calafat Mesquida 2012). Está orientado a procesos, enfocándose en la calidad, proporcionando el mejor servicio posible para cubrir las necesidades de negocio del cliente con los niveles acordados de recursos, lo que hace de él un servicio profesional, rentable y con riesgos asociados que son conocidos y gestionados. Este estándar es aplicable a cualquier organización que se base en servicios de IT, sin importar su tamaño o sector.

ISO/IEC 20000 define un conjunto completo e interrelacionado de procesos de la gestión de servicios. Actualmente, la norma ISO/IEC 20000 se compone de cinco partes, donde tres de ellas han sido publicadas como informe técnico (TR) (Calafat Mesquida 2012):

- ISO/IEC 20000-1:2011: Esta parte define los requisitos para que un proveedor de servicios pueda planificar, establecer, implementar, operar, monitorizar, revisar, mantener y mejorar un sistema de ITSM. Estos requisitos comprenden el diseño, transición, provisión y mejora de los servicios.
- ISO/IEC 20000-2:2012: Proporciona directrices para la implantación de un sistema de ITSM basado en los requisitos de la norma ISO/IEC 20000-1. Incluye ejemplos y sugerencias para interpretar y aplicar la norma ISO/IEC 20000-1.
- ISO/IEC TR 20000-3:2009: Ofrece orientaciones sobre la definición del alcance, la aplicación y la demostración de la conformidad de los proveedores de servicios que pretenden satisfacer los requisitos de la norma ISO/IEC 20000-1.
- ISO/IEC TR 20000-4:2010: Describe un modelo de referencia de procesos de ITSM. Estos procesos son necesarios para satisfacer los requisitos de la norma ISO/IEC 20000-1.
- ISO/IEC TR 20000-5:2010: Ofrece un ejemplo de aproximación incremental a la adopción de la norma ISO/IEC 20000-1 en tres fases secuenciadas.

Entre las ventajas que presenta el uso de la norma ISO 20000 se pueden mencionar:

- Reducción del riesgo ofreciendo apoyo fiable de profesionales de la IT.
- Pone cualquier situación de IT bajo control inmediato y mejora la productividad de los empleados y la fiabilidad del sistema de tecnología de la información.

- La certificación aporta también motivación a la organización, demuestra la fiabilidad y calidad de los servicios de IT para empleados, partes interesadas y clientes.

1.2.3. Integración de Modelos de Madurez de Capacidades para Servicios

CMMI SVC incluye una colección de mejores prácticas para la ITSM, ofreciendo unas directrices para la aplicación del modelo en una organización proveedora de servicios. Las mejores prácticas de CMMI SVC se centran en actividades para proveer servicios de calidad a los clientes y usuarios finales (Calafat Mesquida 2012). Contiene prácticas que cubren los procesos de administración de proyectos, administración de procesos, establecimiento de servicios, entrega de servicios; así como otros procesos de soporte. Está orientado al proceso, pero se enfoca en el desarrollo.

CMMI SVC contiene 24 áreas de proceso, 16 de estas áreas pertenecen al *CMMI Foundation Model* (CMF), 7 pertenecen a la gestión de servicios y el área de proceso restante es compartida con el modelo CMMI DEV (Calafat Mesquida 2012).

La utilización de CMMI SVC presenta múltiples ventajas, las cuales son enunciadas a continuación:

- Direcciona los problemas comunes en los servicios de entrega repetible a través del tiempo, los cambios constantes de clientes y requerimientos.
- Adiciona mejores prácticas enfocadas a servicios, con lo cual se pretende reducir el número de fallas en ellos, mientras se mantiene la disponibilidad de los mismos.
- Mejora la continuidad del servicio y reduce los costos e incidentes de manera efectiva y eficiente; mientras se incrementa la consistencia y la calidad de los servicios proporcionados.
- Presenta una arquitectura de modelo más robusta y con mayor nivel de detalle.
- Es aplicable a más de una disciplina, prestando mejor atención a las áreas de ingeniería.
- La representación continua permite focalizar mejoras de acuerdo a los objetivos del negocio.

1.2.4. Biblioteca de Infraestructura de Tecnologías de Información

ITIL se ha convertido en el estándar conocido y utilizado mundialmente por cientos de organizaciones. Perteneció a la Oficina de Comercio del Gobierno Británico (por sus siglas en inglés OGC), pero es de libre utilización. Es un marco de referencia que describe un conjunto de mejores prácticas y recomendaciones para la administración de servicios de IT, con un enfoque de administración de procesos permitiendo mayor eficiencia en la ITSM, generar orden, lenguaje y procesos comunes que establecen la mejor manera de hacer las cosas (BH Consulting 2005).

ITIL garantiza la reducción de costos por el mejoramiento de los procesos y la reducción de riesgos que afecten el negocio. Otra de sus características es que facilita una mejor integración y relaciones entre IT y el negocio. Es adaptable al cambio y a través del uso de procesos probados mejora los servicios de IT y la satisfacción del cliente a través de la entrega de servicios profesionales. Su principal objetivo es mejorar la calidad de los servicios, evitar los problemas asociados a los mismos y en caso de que ocurra ofrecer un marco de actuación para que sean solucionados con el menor impacto y mayor brevedad posible.

Se organiza en torno a cinco áreas claves (Sallé 2004):

- Perspectiva de negocio.
- Gestión de aplicaciones.
- Prestación de servicios.
- Servicios de apoyo.
- Gestión de infraestructuras.

La utilización de este conjunto de buenas prácticas presenta múltiples ventajas para clientes y usuarios, las cuales son enunciadas a continuación:

- Mejora la comunicación con los clientes y usuarios finales a través de los diversos puntos de contactos acordados.
- Los servicios se definen en lenguaje del cliente y detalladamente.
- Se maneja mejor la calidad y los costos de los servicios.
- La entrega de servicios se enfoca más al cliente, mejorando con ello la calidad de los mismos y relación entre el cliente y el departamento de IT.
- Una mayor flexibilidad y adaptabilidad de los servicios.

1.2.5. Análisis crítico del estándar a utilizar

Se puede concluir diciendo que ITIL es el estándar más factible para el desarrollo del presente trabajo, esto se debe en su totalidad a que este conjunto de buenas prácticas está orientado a procesos, por tanto está presente en el ciclo completo de un servicio. A diferencia de COBIT que está orientado a la gestión de gobierno, es decir a su monitorización y auditoría, enfocándose fuertemente en el control y menos en la ejecución. CMMI SVC se encarga de medir cuando un servicio está listo para ser implantado en una

organización. Por último la norma ISO/IEC 20000, se encarga de medir cuando un servicio tiene la calidad requerida.

Al mismo tiempo el uso de ITIL asegura conseguir una administración de servicios de IT que sea eficiente, efectiva, que esté alineada con las necesidades actuales y futuras de las empresas. También se enfoca en el estudio de mercado y posibilidades mediante la búsqueda de servicios innovadores que satisfagan al cliente tomando en cuenta la real factibilidad de su puesta en marcha, analizándose también posibles mejoras para servicios ya existentes. Es el único estándar que puede integrarse con otros estándares, permitiéndole a una organización alcanzar sus objetivos de negocio.

1.3. Fases del ciclo de vida de la Biblioteca de Infraestructura de Tecnologías de Información

ITIL estructura la ITSM sobre el concepto de Ciclo de Vida de los Servicios, ofreciendo una visión global de la vida de un servicio desde su diseño hasta su eventual abandono, sin ignorar los detalles y funciones de todos los procesos involucrados en la prestación de un servicio. El Ciclo de Vida del Servicio consta de cinco fases que se alimentan entre ellas de una manera cíclica (Long 2011), estas fases se muestran a continuación en la figura 1.



Figura 1: Ciclo de Vida de los Servicios de TI (OSIATIS S.A 2007a).

- **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:** Es la encargada de cubrir 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 través de un diseño, transición y operación del servicio optimizado.

La siguiente figura muestra las 5 fases del Ciclo de Vida de ITIL 2011 con todos los procesos con los que cuenta cada una de ellas.

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 2: Fases y procesos de ITIL 2011 (Corona 2011).

1.4. Procesos de la Biblioteca de Infraestructura de Tecnologías de Información

ITIL 2011 cuenta con 26 procesos en total, es válido destacar dos procesos de suma importancia dentro de la fase de Diseño de Servicio, estos son la Gestión de Catálogo de Servicio y Gestión de Nivel de Servicio.

1.4.1. Proceso Gestión de Niveles de Servicio según la Biblioteca de Infraestructura de la Tecnología de Información

“El proceso de Gestión de Niveles de Servicio perteneciente a la fase de Diseño, asegura la continua identificación, seguimiento y revisión óptima de los niveles acordados de servicios de IT que requiera la empresa. Esto se realiza en estrecha cooperación entre los proveedores de servicios de IT y los clientes” (Sallé 2004).

En este proceso se definen, negocian y supervisan la calidad de los servicios IT ofrecidos, tiene como objetivo velar por la calidad de los servicios IT alineando tecnología con procesos de negocio y todo ello a un coste razonable, además de poner la tecnología al servicio del cliente.

Es imprescindible conocer las necesidades de los clientes, definir correctamente los servicios ofrecidos y monitorizar la calidad del servicio respecto a los objetivos establecidos en los SLAs para poder cumplir con los objetivos principales de la Gestión de Niveles de Servicio.

Es importante agregar que un SLA *“debe recoger en un lenguaje no técnico, o cuando menos comprensible para el cliente, todos los detalles de los servicios brindados”* (OSIATIS S.A 2013). Tras su firma, él debe considerarse el documento de referencia para la relación con el cliente en todo lo que respecta a la provisión de los servicios acordados, por tanto, es imprescindible que contenga claramente definidos los aspectos esenciales del servicio tales como su descripción, disponibilidad, niveles de calidad, tiempos de recuperación, entre otros.

La figura 3 muestra los principales procesos de la Gestión de Niveles de Servicio.



Figura 3: Ciclo de la Gestión de Nivel de Servicio (OSIATIS S.A 2013).

En la planificación se analizan las necesidades del cliente, se elaboran las hojas de especificación del servicio y se establecen los parámetros de rendimiento que permiten verificar la calidad del servicio. La implementación debe establecer en estrecha colaboración con el cliente los SLAs, además de formalizar los OLAs y los UCs con los proveedores externos. Se debe monitorizar la calidad del servicio. Todo proceso debe ser monitorizado asegurando que se cumplen los SLAs, emitiendo informe de rendimiento, elaborando métricas que permitan evaluar los niveles de calidad del servicio. Por último, la revisión, en este proceso se elaboran informes de rendimiento sobre la calidad del servicio, en caso de que fuera necesario se debe modificar los SLAs existentes, además se debe elaborar los planes de mejora del servicio.

1.4.2. Proceso Gestión de Catálogos de Servicio según la Biblioteca de Infraestructura de la Tecnología de Información

Otro proceso importante dentro de la fase de Diseño de servicio es la Gestión de Catálogos de Servicios, anteriormente era un subproceso de la Gestión del Nivel de Servicio, pero se incorporó como un nuevo proceso en ITIL 2011, siendo responsable del diseño de un catálogo de servicios enfocado en las necesidades del cliente.

El Catálogo de Servicio ofrece una descripción detallada de todos los servicios que se prestan y los recursos asignados para ello, sirviéndoles de guía a los clientes para facilitar la selección del servicio que se adapte a sus necesidades. Otra ventaja es que delimita las funciones y compromisos de la organización IT, puede ser utilizado como una herramienta de venta, evitando malentendidos entre los diferentes actores implicados en la prestación de un servicio. Es imprescindible la alineación de los aspectos técnicos con políticas de negocio para lograr la realización del catálogo de servicio.

La Gestión de Catálogos está resumida en las siguientes actividades (OSIATIS S.A 2007b):

- **Definición de servicios:** Esta actividad consiste en trazar las líneas de servicios o familias principales en las que se van a agrupar. Generalmente, las familias de servicios están relacionadas con las áreas funcionales en las que se desarrollan estas. Seguidamente se van detallando los servicios existentes en cada una de las familias de servicios, así como los clientes que los han contratado y la demanda prevista para cada servicio.
- **Mantenimiento y actualización:** Consiste en definir los destinatarios y el propósito de la información detallada en el catálogo. También se debe planificar las tareas de actualización de la información consignada en el catálogo de servicios. Además de programar revisiones periódicas, deben estipularse de antemano los casos que pueden requerir una actualización extraordinaria y los protocolos para la aprobación de estos cambios.

La siguiente figura muestra las interacciones y funcionalidades del Catálogo de Servicios.



Figura 4: Interacciones y funcionalidades del Catálogo de Servicios (OSIATIS S.A 2007b).

El proceso de Gestión de Catálogos de Servicios debe estar interrelacionado con otros procesos como la Gestión Financiera, Gestión de Niveles de Servicio, Gestión de Proveedores y el Centro de Servicios para poder aportar a todas las organizaciones IT información sobre los servicios activos y poder fundamentar la relación con los clientes tanto desde el punto de vista comercial como en el de soporte técnico. También es importante llevar a cabo la monitorización para asegurar que la información del Portafolio de Servicios esté actualizada, el resto de los procesos de IT estén informados y que el personal de la organización hace uso del Catálogo de Servicios de forma habitual.

Se debe tomar la información generada por el proceso de Gestión del Portafolio de Servicios para complementarla con otros documentos de interés originados en este proceso. Seguidamente en la fase de Definición se crean una serie de familias de servicios para agrupar los servicios por afinidad y registrar la información relativa a cada servicio. Finalmente en la fase de Mantenimiento y actualización se establecen una serie de pautas de consulta, se forma al personal de la organización, se planifican las tareas de mantenimiento y se tipifican los casos para actuaciones extraordinarias.

El proceso de Gestión de Catálogos de Servicios debe estar interrelacionado con otros procesos como la Gestión Financiera, Gestión de Niveles de Servicio, Gestión de Proveedores y el Centro de Servicios para poder aportar a todas las organizaciones IT información sobre los servicios activos y poder fundamentar la relación con los clientes tanto desde el punto de vista comercial como en el de soporte técnico. También es importante llevar a cabo la monitorización para asegurar que la información del Portafolio de Servicios esté actualizada, el resto de los procesos de IT estén informados y que el personal de la organización

hace uso del Catálogo de Servicios de forma habitual.

Se debe tomar la información generada por el proceso de Gestión del Portafolio de Servicios para complementarla con otros documentos de interés originados en este proceso. Seguidamente en la fase de Definición se crean una serie de familias de servicios para agrupar los servicios por afinidad y por último se registra la información relativa a cada servicio. Finalmente en la fase de Mantenimiento y actualización se establecen una serie de pautas de consulta, se forma al personal de la organización, se planifican las tareas de mantenimiento y se tipifican los casos para actuaciones extraordinarias.

1.5. Ventajas de la Biblioteca de Infraestructura de Tecnologías de Información para la Tecnología de Información

- La organización IT desarrolla una estructura más clara, se vuelve más eficaz y se centra más en los objetivos de la organización.
- La administración tiene un mayor control, se estandarizan e identifican los procedimientos y los cambios resultan más fáciles de manejar.
- La estructura de procesos en IT proporciona un marco para concretar de manera más adecuada los servicios de *outsourcing*.
- A través de las mejores prácticas de ITIL se apoya al cambio en la cultura de IT y su orientación hacia el servicio y se facilita la introducción de un sistema de administración de calidad.
- ITIL proporciona un marco de referencia uniforme para la comunicación interna y con proveedores.

1.6. Ventajas al adoptar la Biblioteca de Infraestructura de Tecnologías de Información para la Tecnología de Información

ITIL brinda los cimientos de la Gestión de Servicios Informáticos. Apoya activamente los objetivos empresariales ofreciendo servicios que se basan en principios eficientes y que cumplen adecuadamente los requisitos del negocio. Puede convertirse en un generador de beneficios en vez de considerarse como una carga costosa. Las múltiples ventajas que se obtienen al adoptar ITIL se enuncian a continuación (Axios Systems 2011):

- Alinea los servicios informáticos a las necesidades actuales y futuras de los negocios y sus clientes.
- Mayor accesibilidad a servicios para los usuarios mediante un único punto de contacto.
- Respuestas rápidas a las consultas y reclamaciones de clientes.

- Mejor trabajo en equipo y comunicación.
- Percepción más favorable de los servicios prestados.
- Mejor calidad de información de carácter informático para la gestión y toma de decisiones óptimas.
- Mejor gestión y control de la infraestructura del sistema informático.
- Uso más eficaz y eficiente de los recursos correspondientes a la prestación de servicios con el consiguiente potencial de reducción de costes.
- Mejor atención y mayor satisfacción del cliente.
- Mayor control sobre el rendimiento en cuanto a los SLAs.
- Eliminación de pérdidas y falta de homogeneidad en los registros de información, incidentes y consultas de clientes.
- Reducción en el número de incidentes.

1.6.1. Integración de estándares

ITIL no es un método, sino un conjunto de buenas prácticas, lo que constituye otra ventaja, esto se debe en su totalidad, a la facilidad de integrarlo con otros marcos de referencia, buenas prácticas y métodos. De esta manera se logra que una organización alcance sus objetivos de negocio. Cada marco de referencia, buenas prácticas y métodos presentan características diferentes, porque fueron creados por entes y con propósitos distintos. Es necesario saber cuál es el que se ajusta a una determinada problemática, porque cada uno presenta un enfoque específico y un nivel de granularidad distinto. A continuación se especifican las principales características de integrar el conjunto de buenas prácticas ITIL con COBIT, ISO/IEC 20000 y CMMI SVC respectivamente.

Integración de la Biblioteca de Infraestructura de la Tecnología de la Información con el estándar Objetivos de Control para la Información y la Tecnología

Ambos modelos son complementarios y se pueden usar juntos, ITIL para lograr efectividad y eficiencia en los servicios IT y 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 (Coello 2008). COBIT tiene mayor alcance que ITIL ya que abarca todo el espectro de actividades de IT, mientras que ITIL está centrado solo en Gestión de Servicio.

Integración de la Biblioteca de Infraestructura de la Tecnología de la Información con el estándar Organización Internacional de Normalización/Comisión Electrotécnica Internacional 20000

La norma ISO/IEC 20000 es totalmente compatible con ITIL, la diferencia es que ITIL no es medible y puede ser implantado de muchas maneras, mientras que en la ISO/IEC 20000, las organizaciones deben ser auditadas y medidas frente a un conjunto establecido de requisitos. Es válido agregar que la certificación de esta norma está basada en ITIL (Turbitt 2006).

Integración de la Biblioteca de Infraestructura de la Tecnología de la Información con el estándar Integración de Modelos de Madurez de Capacidades para Servicios

Ambos modelos son complementarios. CMMI se centra en garantizar la calidad en el desarrollo de *software* mientras que ITIL garantiza la explotación del producto de *software*, por ello ambas metodologías no son excluyentes, utilizándose en proyectos de análisis y definición de procesos que permitan ajustar ambas filosofías de trabajo, abarcan desde el desarrollo del *software* hasta la gestión del mantenimiento y servicio del mismo (Coello 2008).

1.7. Análisis de soluciones existentes

Antes de comenzar con la implementación de la solución propuesta, se realizó un análisis del estado del arte permitiendo el estudio de algunos sistemas existentes en el mundo y en Cuba que gestionan catálogos, niveles de servicios y cumplen con el conjunto de buenas prácticas ITIL. Este estudio permitió obtener las principales características y ventajas de estos sistemas, posibilitando que la implementación de la solución propuesta de respuesta al objetivo trazado. Según los indicadores y la búsqueda realizada se seleccionaron estos sistemas por contar con suficiente documentación disponible para su estudio. A continuación se detallan algunas de las particularidades fundamentales de las soluciones analizadas.

1.7.1. Remedy IT Service Management

Este *software* permite que los usuarios sean capaces de implementar flujos de trabajo ITIL pre-integrados para alinear proactivamente las operaciones de IT con los requerimientos del negocio. Favorece la reducción de costes y una importante mejora en los niveles de servicios. Incluye cuatro aplicaciones, en las que se destaca *BMC Service Level Management*, la misma proporciona una cobertura comprensible y sencilla del proceso de gestión de niveles de servicio, facilitando al usuario moverse desde métricas de gestión tecnológicas puramente técnicas a otras más centradas en el negocio (BMC Software 2005).

1.7.2. HP Service Manager

Este sistema cumple con el estándar ITIL v3, presenta una arquitectura modular la cual permite partir de una implantación pequeña ir añadiendo módulos, que van a permitir incrementar las funcionalidades adaptándose a las necesidades del cliente. Entre los módulos que presenta, se destacan el módulo Gestión de niveles de servicio, el cual está diseñado para ayudar a mantener servicios de IT alineados con las necesidades empresariales, permite desarrollar los objetivos de nivel de servicio para llamadas, solicitudes de usuario, incidencias, problemas y cambios. Otro módulo es Catálogo de servicios, el cual facilita la gestión de *leasings*, mantenimientos, garantías y licencias de *software* de los activos (HP Software 2011).

1.7.3. IBM Tivoli Service Request Manager

Este producto cumple con los procesos de ITIL v3, ofrece una solución unificada y fluida para manejar todos los aspectos de las peticiones de servicio. Combina las funciones de centro de servicio al usuario y de catálogo de servicios con un motor de automatización de procesos común. De este modo, se crea un sistema de gestión de peticiones automatizado apoyado por un proceso optimizado de soporte y entrega. Permite a los usuarios seleccionar directamente los servicios desde un catálogo visual intuitivo, lo que permite reducir costes. Este catálogo personalizado también refleja los términos de los acuerdos de nivel de servicio, los acuerdos contractuales y los términos de valoración y facturación. Además agiliza los procesos de gestión de peticiones basadas en ITIL, incidentes, problemas, nivel de servicio, conocimiento, catálogo de servicios y gestión financiera (IBM España S.A 2011).

1.7.4. ManageEngine ServiceDesk Plus

ManageEngine ServiceDesk Plus es el encargado de prestar servicios de soporte técnico a las aplicaciones informáticas y productos desarrollados por cada centro productivo de la Universidad. Brinda servicios de acceso a noticias, artículos y eventos relacionados con aplicaciones informáticas, red de centros de desarrollo de la UCI y avances tecnológicos; descarga de ambientes de desarrollo y aplicaciones informáticas de código abierto desarrolladas por la red de centros de desarrollo de la UCI; documentación en línea de soluciones informáticas adquiridas, actualizaciones, base de conocimientos y gestión de incidencias.

1.7.5. Análisis crítico de las soluciones estudiadas

Los sistemas internacionales analizados cumplen con los procesos de ITIL v3, pero este conjunto de buenas prácticas sufrió cambios en el 2009 debido a errores presentados, para corregirlos se hicieron varias modificaciones a lo largo de los cinco libros que representan el ciclo de mejora continua del

servicio, creándose una nueva edición, ITIL 2011. Por tanto, estos sistemas no se adaptan correctamente a los procesos de gestión de catálogos y gestión de nivel de servicios. Además estos sistemas utilizan licencias privativas, lo cual constituye una desventaja para Cuba porque requiere de una compra del sistema para su posterior utilización y por consiguiente un costo adicional para su mantenimiento.

Luego de analizar la solución que se utiliza en la UCI, se llega a la conclusión, que es de suma importancia el desarrollo de un subsistema para la Gestión de Catálogos y Gestión de Nivel de servicio para el Centro de Soporte UCI, porque el sistema usado en este centro es una herramienta propietaria que impide el mantenimiento de sus funcionalidades y que no posibilita que se actualice la forma en que se realiza el trabajo con ella, atendiendo a las particularidades del centro y la evolución en lo que a servicio de soporte técnico respecta.

1.8. Metodología de desarrollo de software

En el proceso de desarrollo de *software* es necesaria la utilización de un elemento fundamental, una metodología de desarrollo de *software*, la misma constituye “*una filosofía de desarrollo de programas de computación acompañada de herramientas, modelos y métodos, que es usada para estructurar, planear y controlar el proceso de desarrollo de un software*” (Beck 2004). Definiendo quién, qué, cuándo y cómo debe hacerlo.

Las metodologías de desarrollo de *software* se clasifican en metodologías pesadas y metodologías ágiles. El éxito de un producto depende grandemente de la metodología escogida por el equipo de desarrollo, sin importar su clasificación, lo significativo es que aumente la calidad del producto final con los recursos y el tiempo establecido.

Metodologías pesadas: Se centran en documentación, planificación y procesos. Cierta resistencia a los cambios, más artefactos, más roles. El cliente interactúa con el equipo de desarrollo mediante reuniones. Este enfoque tradicional es efectivo y necesario en proyectos de gran tamaño respecto a tiempo y recursos.

Metodologías ligeras o ágiles: A partir de la década del 90 se inicia el desarrollo de metodologías con una filosofía de desarrollo diferente a las vistas hasta ese momento, esto se debe en su totalidad al entorno cambiante del sistema en los proyectos actuales, donde se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Debido a estas restricciones de flexibilidad y tiempo se dificulta la utilización de las metodologías pesadas que se utilizaban hasta ese momento. Pero con el surgimiento de las metodologías ligeras o ágiles se le dio solución a los problemas antes mencionados, las mismas constituyen un nuevo enfoque en el desarrollo de *software*, mejor aceptado por

los desarrolladores de proyectos que las metodologías pesadas o convencionales debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración. Otras de sus características es que genera pocos artefactos, pocos roles y el cliente es parte del equipo de desarrollo. Entre las metodologías ágiles existentes, se destacan SCRUM, *Crystal Methodologies*, *Adaptive Software Development* y Programación Extrema (por sus siglas en inglés XP).

Para guiar el proceso de desarrollo de *software* del presente trabajo, los autores seleccionaron la metodología XP⁴, aunque constituye una de las metodologías ágiles más usadas es la que más se adecua a las características del equipo de desarrollo, debido a que está orientada especialmente para proyectos pequeños, genera pocos artefactos y poca documentación, lo cual es una ventaja para los proyectos con un corto plazo de entrega. Centrada en potenciar las relaciones interpersonales, el trabajo en equipo y el aprendizaje de los desarrolladores. *“Además se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, elemento clave para el éxito en desarrollo de software, la simplicidad en las soluciones implementadas y la reutilización del código desarrollado”* (Fernández Escribano 2002). El uso de la misma está definido especialmente para proyectos con requisitos imprecisos y muy cambiantes. Su particularidad consiste en tener al usuario final como parte del equipo de desarrollo. *“El ciclo de vida ideal de esta metodología consta de seis fases”* (Remedios Cruz 2012):

- **Exploración:** Los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Además el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto.
- **Planificación de la Entrega:** El cliente establece la prioridad de cada Historia de Usuario (HU) y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas.
- **Iteraciones:** Incluye varias iteraciones sobre el sistema antes de ser entregado.
- **Producción:** Requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente.
- **Mantenimiento:** Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones.
- **Muerte del Proyecto:** Es cuando el cliente no tiene más HU para ser incluidas en el sistema. También ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no

⁴ Una metodología ágil de desarrollo de *software* que incluye prácticas tales como escenario basado en requisitos y programación en pares.

hay presupuesto para mantenerlo.

1.9. Herramientas y tecnologías utilizadas

En el presente epígrafe se describen las principales características de las herramientas y tecnologías necesarias para el desarrollo del presente trabajo. Las mismas fueron seleccionadas porque la presente investigación es parte de un trabajo de doctorado, el cual tiene como propósito la integración de varios módulos para conformar una plataforma de Gestión de Servicio, por tanto es necesario la selección de las mismas herramientas y tecnologías de los trabajos investigativos desarrollados anteriormente para garantizar la compatibilidad de estos sistemas.

1.9.1. Groovy and Grails

Como marco de trabajo se utilizará Grails en su versión 2.1.1. El mismo es un *framework*⁵ libre para aplicaciones web, desarrollado sobre el lenguaje de programación Groovy (el cual a su vez está diseñado para correr sobre la máquina virtual de Java).

Grails pretende ser un *framework* altamente productivo siguiendo paradigmas tales como convención sobre configuración o no te repitas (DRY), proporcionando un entorno de desarrollo estandarizado y ocultando gran parte de los detalles de configuración al programador. Tiene tres características que intentan incrementar su productividad comparándolo con los *framework* Java tradicionales, inexistencia de configuración XML⁶, entorno de desarrollo preparado para funcionar desde el primer momento y funcionalidad disponible mediante métodos dinámicos, está construido sobre la plataforma Java, con lo que es muy fácil integrarlo con librerías Java, *framework* y códigos existentes (Brito 2009a).

Groovy es un lenguaje relativamente nuevo, puede ser interpretado o compilado y está diseñado específicamente para la plataforma Java. No sólo tiene acceso a la API⁷ de Java existente; se extiende a la API de Java mediante la adición de nuevos métodos a las clases existentes. Groovy ha demostrado ser una gran plataforma para conceptos tales como metaprogramación y dominio específico de idiomas (Judd, Nusairat and Shingler 2008).

1.9.2. IntelliJ IDEA

Como IDE⁸ se utilizará IntelliJ IDEA en su versión 12.1. El mismo se autodefine como un entorno inteligente para desarrollar aplicaciones Java, cliente y servidor. Su uso permite escritura de código sin

⁵ Marco de trabajo.

⁶ Lenguaje de Marcado Extensible.

⁷ Interfaz para Programación de Aplicaciones.

⁸ Entorno de Desarrollo Integrado.

complicaciones, practica un abordaje no intrusivo e intuitivo para ayudar a escribir, depurar, refactorizar, probar y aprender el código; valida la calidad de este y ofrece soluciones inmediatas para los problemas encontrados en todos los niveles desde la instrucción individual para arquitectura global, utilizando las inspecciones de código avanzado y análisis de matriz de dependencia. IntelliJ IDEA puede coexistir con otros IDEs populares, como Eclipse y herramientas de gestión de proyectos como Maven, para que su equipo pueda utilizar cada herramienta donde es mejor aplicable (Brito 2009b).

1.9.3. Apache Tomcat

Como servidor web se utilizará Apache Tomcat en su versión 7.0, este es un servidor web con soporte de *servlets*⁹ y JSPs¹⁰, incluye el compilador Jasper, que compila JSPs convirtiéndolas en *servlets*. El motor de *servlets* de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad (Apache Software Foundation 2014).

1.9.4. PostgreSQL

Para la gestión de base de datos se utilizará PostgreSQL en su versión 9.1. El mismo es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (Martínez 2009).

1.9.5. Bootstrap

Bootstrap es una colección de herramientas de *software* libre para la creación de sitios y aplicaciones web. Contiene plantillas de diseño basadas en HTML¹¹ y CSS¹² 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

⁹ Es una clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor.

¹⁰ Páginas de Servidor de Java.

¹¹ Lenguaje de Marcado de Hipertexto.

¹² Hojas de Estilo en Cascada.

fomentar la consistencia a través de herramientas internas. Es de código abierto, además de ser compatible con la mayoría de los navegadores web (Ramírez 2013). Se utilizará la versión 2.3.2.

1.9.6. Visual Paradigm

Para el modelado se utilizará Visual Paradigm en su versión 5.0. Es una herramienta CASE (en español Ingeniería de *Software* Asistida por Computadora)¹³, para desarrollo de aplicaciones. Está concebida para apoyar el ciclo de vida completo del proceso de desarrollo de *software*, a través de la representación de diagramas. Posee una distribución automática de diagramas, contando con una reorganización de las figuras y conectores de los diagramas UML¹⁴. Captura requisitos mediante el modelado de los casos de uso. Permite además exportar los diagramas a imágenes y páginas HTML (Visual Paradigm 2010).

1.9.7. JMeter

Para la realización de las pruebas de rendimiento se utilizó la herramienta JMeter en su versión 2.9. Esta herramienta está diseñada para realizar pruebas de rendimiento y pruebas funcionales sobre aplicaciones web. Permite la ejecución de pruebas distribuidas entre distintos ordenadores. Tiene la forma de generar un caso de prueba a través de una navegación de usuario. Además permite la simulación de una amplia gama de recursos de *software* en tiempo de ejecución real, a partir de solicitudes sencillas hasta otras de mayor envergadura (Apache Software Foundation, 2013).

1.10. Conclusiones parciales

- El estudio de los conceptos fundamentales permitió un mayor entendimiento de la problemática a resolver y la identificación de las tendencias actuales y particularidades de los procesos de Gestión de Catálogos y Gestión de Nivel de Servicio.
- La utilización del conjunto de buenas prácticas ITIL proporcionó un flujo de trabajo para el desarrollo de los procesos Gestión de Catálogos y Gestión de Nivel de Servicio.
- Con el análisis de las principales características y funcionalidades de los sistemas homólogos estudiados se corroboró la necesidad de implementar un subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio para contribuir al correcto funcionamiento de estos procesos en el Centro de Soporte.
- El estudio de la metodología XP posibilitó la identificación de las fases y los artefactos que se generan en cada una de ellas.

¹³ Sistemas de *software* destinados a proporcionar soporte automatizado para las actividades del proceso de *software*.

¹⁴ Lenguaje de Modelado Unificado.

ANÁLISIS Y DISEÑO DEL SUBSISTEMA DE GESTIÓN DE CATÁLOGOS Y GESTIÓN DE NIVEL DE SERVICIO



Es prioridad para el Centro de Soporte de la UCI el desarrollo de un sistema que contribuya a la realización de los procesos de Gestión de Catálogos y Gestión de Niveles de Servicio, para satisfacer las necesidades de los clientes. Esto es una tarea compleja, pero al llevar a cabo el proceso de análisis y diseño del sistema se proporciona una guía útil para disminuir las situaciones de fracasos o errores.

En el presente capítulo se presentan, las principales características y funcionalidades de la solución propuesta. Se definirán los requisitos funcionales y no funcionales basándose en las necesidades del cliente y en las restricciones del sistema.

2.1 Solución propuesta

El Centro de Soporte UCI es el principal encargado de brindarle servicios de soporte técnico a todos los sistemas desarrollados en diferentes centros productivos. Para ello cuenta con un sistema, en el cual se evidencian todos los servicios prestados, la descripción de los mismos, los responsables, entre otros datos de sumo interés. Esta plataforma usada es propietaria e impide la actualización de servicios, por tanto el Catálogo de Servicio está desactualizado imposibilitándole a los clientes conocer todos los servicios con los que cuenta el centro, además la Gestión de Niveles de Servicio está incompleta, existen diversos problemas en la documentación que este proceso genera, porque no se encuentran informatizadas, lo que impide su posterior consulta.

Por los problemas existentes en el Centro de Soporte UCI y para darle solución a los mismos se decide la implementación de un subsistema de Gestión de Catálogos de Servicios y Gestión de Niveles de Servicio (GESERV) para lograr contribuir al correcto funcionamiento de estos procesos en el Centro de Soporte UCI.

Mediante un catálogo, el cliente podrá seleccionar los servicios que se ajusten a sus necesidades, garantizando la calidad de los servicios prestados teniéndose en cuenta los SLAs establecidos por el mismo. Además se permitirá la creación y posterior monitorización de estos acuerdos teniendo en cuenta los tiempos de respuestas y las resoluciones establecidas. Mediante el uso de este subsistema se podrá establecer de manera automática los SLAs, OLAs y UCs que se aplicarán a cada petición, comprobándose posteriormente su grado de cumplimiento. Por consiguiente, se verificará si los servicios se prestan en plazo o si están próximos a caducar, para así priorizar todas las peticiones pendientes. También se mostrarán reportes gráficos con la cantidad de SLAs, OLAs, Servicios, Áreas funcionales y

Proveedores existentes en la base de datos. Para acceder al sistema, se debe poseer una cuenta de usuario, el encargado de gestionar estas cuentas es el Administrador, además de gestionar toda la información con la que cuenta el sistema y de asignarle roles a los usuarios. Estos roles pueden ser Administrador, Especialista, Técnico o Invitado, cada rol tiene diferentes responsabilidades y niveles de acceso. Es válido agregar que cuando una nueva cuenta de usuario es creada o cuando se asigna uno o más usuarios a un grupo determinado, el sistema envía notificaciones por correo personales o grupales.

2.2 Actores del sistema

En la tabla 1 se muestran los disímiles roles que puede tener un usuario en el Subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio, así como una breve descripción de sus responsabilidades y niveles de acceso.

Rol	Descripción
Administrador	Tiene acceso a todas las vistas y funcionalidades de la solución propuesta. Es el encargado de gestionar las cuentas, roles y grupos de usuarios; además de gestionar los acuerdos de nivel de servicio, de operación, áreas funcionales, servicios, proveedores, impactos, urgencias, evaluaciones y departamentos.
Especialista	Tiene acceso a todas las vistas y funcionalidades de la solución propuesta, excepto a las relacionadas con las cuentas de usuario. Es el encargado de gestionar los acuerdos de nivel de servicio, de operación, áreas funcionales, servicios, proveedores, impactos, urgencias, evaluaciones y departamentos.
Técnico	No puede acceder a las vistas y funcionalidades relacionadas con las cuentas de usuario y nomencladores. Es el encargado de gestionar los acuerdos de nivel de servicio, de operación, áreas funcionales, servicios, proveedores y departamentos.
Invitado	Puede acceder a las vistas relacionadas con la gestión del catálogo de servicios, matriz de asignación de prioridad, gráfico de monitoreo, últimos acuerdos de niveles de servicios y operaciones. Además puede realizar búsqueda simple y avanzada.

Tabla 1: Actores del sistema.

2.3 Modelo conceptual

Al realizar un análisis de los problemas existentes en el Centro de Soporte UCI, se determinó que todos los procesos no están bien definidos. Por esta razón, es necesaria la realización de un modelo conceptual para identificar algunas de las clases que se utilizarán en la solución propuesta.

Un modelo conceptual permite una mayor comprensión del proceso de negocio, a través de la representación de un conjunto de conceptos claves del dominio del problema, para verificar y validar la comprensión del mismo entre las diversas partes involucradas. Es el artefacto más importante que se crea durante el análisis orientado a objetos. Con este modelo no se debe representar componentes o artefactos de *software*.

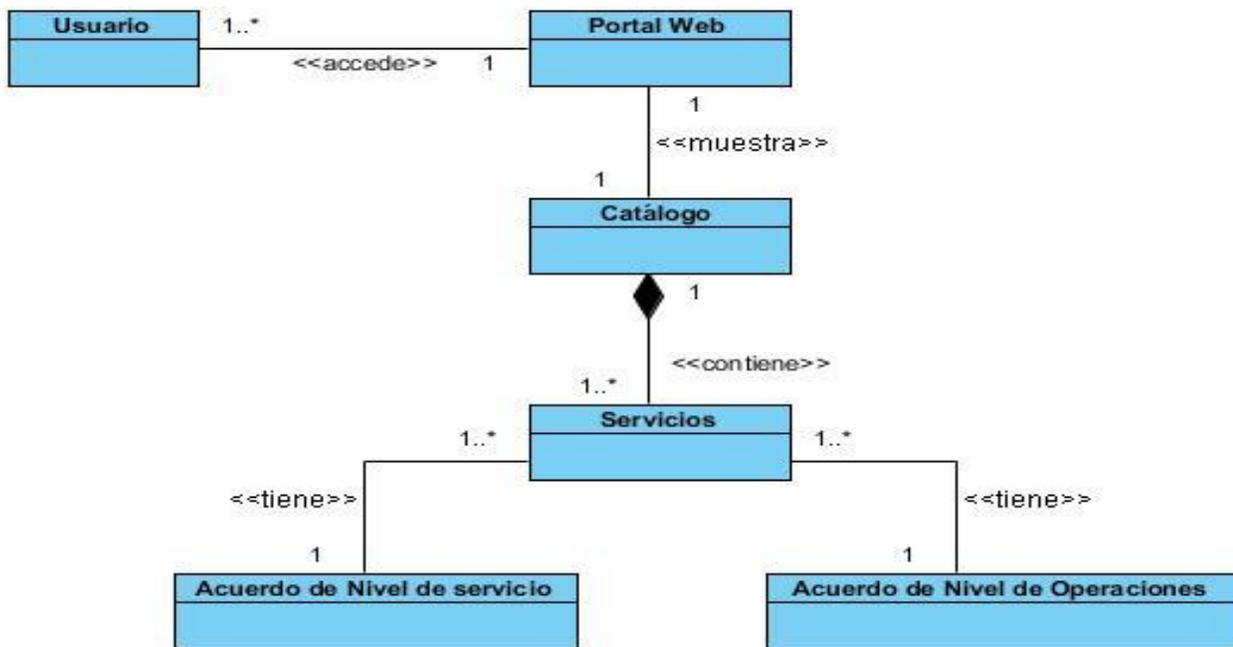


Figura 5: Modelo Conceptual.

A continuación se enuncian un conjunto de conceptos que conforman la terminología del modelo conceptual para facilitar un mayor entendimiento de los términos tratados en el diagrama anterior.

Portal Web: Contiene todos los servicios con los que cuenta el Centro de Soporte.

Usuario: Representa todos los tipos de usuarios existentes en el sistema.

Catálogo: Interfaz que muestra todos los servicios con los que cuenta el sistema.

Servicio: Representa los servicios de soporte técnico que ofrecerá el Centro de Soporte UCI.

Acuerdo de Nivel de Servicio: Establece todas las condiciones que debe cumplir un servicio determinado.

Acuerdo de Nivel de Operación: Establece las responsabilidades y compromisos internos de la organización.

2.4 Requisitos del sistema

En el proceso de desarrollo de *software* es imprescindible la captura de requisitos y está extensamente reconocido dentro de la industria del *software* que los proyectos son críticamente vulnerables cuando estas actividades se realizan mal. Los requisitos de un sistema son la descripción de los servicios y de sus limitaciones operativas. Representan las características, restricciones y funcionalidades que debe tener un sistema. Estos requisitos reflejan las necesidades de los clientes en contar con un sistema que ayude a resolver determinados problemas (Sommerville 2006).

2.4.1 Requisitos funcionales

Los requisitos funcionales (RF) son declaraciones de los servicios que los sistemas deben proporcionar, cómo el sistema debe reaccionar a entradas particulares y cómo debe comportarse en situaciones determinadas. En otros casos, los requisitos funcionales también pueden declarar explícitamente lo que el sistema no debe hacer (Sommerville 2006). A continuación se exponen los requisitos funcionales del sistema:

RF1: Autenticar usuario.

RF2: Gestionar usuario.

RF2.1: Insertar usuario.

RF2.2: Modificar usuario.

RF2.3: Eliminar usuario.

RF2.4: Ver usuario.

RF3: Gestionar grupo de usuario.

RF3.1: Insertar grupo de usuario.

RF3.2: Modificar grupo de usuario.

RF3.3: Eliminar grupo de usuario.

RF3.4: Ver grupo de usuario.

RF4: Gestionar rol de usuario.

RF4.1: Insertar rol de usuario.

RF4.2: Modificar rol de usuario.

RF4.3: Eliminar rol de usuario.

RF4.4: Ver rol de usuario.

RF5: Gestionar acuerdo de nivel de servicio.

RF5.1: Insertar acuerdo de nivel de servicio

RF5.2: Modificar acuerdo de nivel de servicio

RF5.3: Eliminar acuerdo de nivel de servicio.

RF5.4 Ver acuerdo de nivel de servicio.

RF6: Gestionar servicio.

RF6.1: Insertar servicio.

RF6.2: Modificar servicio.

RF6.3: Eliminar servicio.

RF6.4: Ver servicio.

RF7: Gestionar área funcional.

RF7.1: Insertar área funcional.

RF7.2: Modificar área funcional.

RF7.3: Eliminar área funcional.

RF7.4: Ver área funcional.

RF8: Gestionar acuerdo de nivel de operación

RF8.1: Insertar acuerdos de nivel de operación.

RF8.2: Modificar acuerdo de nivel de operación.

RF8.3: Eliminar acuerdo de nivel de operación.

RF8.4: Ver acuerdo de nivel de operación.

RF9: Gestionar proveedor.

RF9.1: Insertar proveedor.

RF9.2: Modificar proveedor.

RF9.3: Eliminar proveedor.

RF9.4: Ver proveedor.

RF10: Mostrar catálogo de servicios.

RF10.1: Insertar catálogo de servicios.

RF10.2: Modificar catálogo de servicios.

RF10.3: Eliminar catálogo de servicios.

RF10.4: Ver catálogo de servicios.

RF11: Gestionar evaluación.

RF11.1: Insertar evaluación.

RF11.2: Modificar evaluación.

RF11.3: Eliminar evaluación.

RF11.4: Ver evaluación.

RF12: Gestionar urgencia.

RF12.1: Insertar urgencia.

RF12.2: Modificar urgencia.

RF12.3: Eliminar urgencia.

RF12.4: Ver urgencia.

RF13: Gestionar Impacto.

RF13.1: Insertar impacto.

RF13.2: Modificar impacto.

RF13.3: Eliminar impacto.

RF13.4: Ver impacto.

RF14: Gestionar prioridad.

RF14.1: Insertar prioridad.

RF14.2: Modificar prioridad.

RF14.3: Eliminar prioridad.

RF14.4: Ver prioridad.

RF15: Gestionar departamento.

RF15.1: Insertar departamento.

RF15.2: Modificar departamento.

RF15.3: Eliminar departamento.

RF15.4: Ver departamento.

RF16: Mostrar reportes gráficos con la cantidad de acuerdos de niveles de servicios, acuerdos de niveles de operación, servicios, áreas funcionales de los servicios y proveedores existentes.

RF17: Asociar a un servicio un acuerdo de nivel de servicio existente.

RF18: Asociar un servicio a un acuerdo de nivel de operación.

RF19: Asociar a un servicio una o más áreas.

RF20: Realizar búsqueda simple.

RF21: Realizar búsqueda avanzada.

RF22: Realizar notificaciones por correo a los usuarios.

RF23: Realizar notificaciones por correo a un grupo de usuarios.

RF24: Asociar una evaluación a los proveedores.

RF25: Realizar reportes de los acuerdos de niveles de servicios y acuerdos de niveles de operación en diferentes formatos.

RF26: Adjuntar documento (archivo) a los acuerdos de niveles de servicios existentes.

RF27: Adjuntar documento (archivo) a los servicios existentes.

RF28: Mostrar los últimos acuerdos de niveles de servicio tomados con el cliente.

RF39: Mostrar los últimos acuerdos de niveles de operación pactados por el equipo de trabajo.

RF30: Mostrar matriz de asignación de prioridad.

RF31: Mostrar información del sitio (manual de ayuda).

2.4.2 Requisitos no funcionales

Los requisitos no funcionales (RN) especifican propiedades o cualidades que el producto de *software* debe tener, como restricciones en cuanto al diseño, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, seguridad, entre otras. Estos requisitos son muy importantes ya que el sistema tiene que ser agradable, sencillo, fiable, es decir que contenga elementos atractivos para los usuarios (Sommerville 2006). A continuación se exponen los requisitos no funcionales del sistema:

Usabilidad

RNU 1: El sistema debe presentar un acceso fácil y rápido, para facilitar el uso del mismo por usuarios con pocos conocimientos en el campo de la informática.

RNU 2: Después de una formación de 2 horas a los usuarios experimentados les deberá ser posible utilizar todas las funciones del sistema.

RNU 3: La interfaz debe ser sencilla y amigable de manera que potencie la comodidad del usuario para su trabajo, además de que las opciones más usadas presentarán vías rápidas y cómodas de invocarse.

Diseño e implementación

RNDI 1: Se utilizará PostgreSQL en su versión 9.1 o superior como Sistema Gestor de Bases de Datos.

RNDI 2: Se utilizará Groovy como lenguaje de programación y Grails como framework integrador.

RNDI 3: Se utilizará como arquitectura del sistema Modelo Vista Controlador (MVC).

RNDI 4: Se utilizará IntelliJ Idea como IDE para el desarrollo del sistema.

RNDI 5: Se utilizará Visual Paradigm como herramienta de modelado.

RNDI 6: Para el diseño de las páginas se utilizarán las hojas de estilo en cascadas (CSS por sus siglas en inglés), lenguaje de marcas (o etiquetas) hipertexto (HTML por sus siglas en inglés), lenguaje de marcas hipertexto extensible (XHTML por sus siglas en inglés), JavaScript y JQuery.

Funcionamiento

Hardware:

RNFO 1: El servidor para aplicaciones web deberá tener las siguientes características mínimas:

- 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.

RNFO 2: El servidor para la base de datos deberá tener las siguientes características mínimas:

- 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.

RNFO 3: Las estaciones de trabajo deberán cumplir los siguientes requisitos mínimos:

- Procesador Intel Celeron E3200 a 2.40 GHz, equivalente o superior.
- 256 Mb de memoria RAM.

Seguridad

Confidencialidad:

RNS 1: El acceso al sistema, así como la información se encontrarán protegidos contra accesos no autorizados utilizando mecanismos de autenticación propios del sistema.

RNS 2: Las diferentes áreas del sistema se encontrarán protegidas contra acceso no autorizado utilizando roles y grupos de usuarios.

Integridad:

RNS 3: La información podrá ser modificada solo por el personal autorizado.

RNS 4: Se harán validaciones de la información en el servidor contra ataques de inyección HTML y SQL.

RNS 5: Se protegerá la información contra ataques CSRF.

Fiabilidad

RNF 1: La respuesta del sistema ante una búsqueda bajo criterios no debe exceder los 7 segundos.

RNF 2: Ante una inserción de datos, vista de detalles, modificación o eliminación el sistema debe responder en no más de 3 segundos.

RNF 3: El sistema debe responder en un promedio de 6 segundos la petición de un reporte.

RNF 4: El sistema deberá soportar transacciones de cerca de 300 usuarios conectados simultáneamente.

Interfaz de usuario

RNIU 1: Las interfaces del sistema contendrán los datos de forma estructurada, permitiendo la interpretación correcta de la información.

RNIU 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.

RNIU 3: El diseño de la interfaz del sistema responderá a la ejecución de acciones de forma rápida, minimizando los pasos a dar en cada proceso.

Los datos más específicos se recogen en la Lista de Reserva del Producto. Esta lista se encuentra reflejada en el [Anexo 4](#).

2.5 Fase de exploración

Esta fase toma pocas semanas o pocos meses en dependencia de la familiarización de los desarrolladores con la tecnología. A grandes rasgos se plantean las HU que son de sumo interés para la primera entrega del producto, además el equipo de desarrollo debe conocer las herramientas, tecnologías y prácticas que se utilizarán para el desarrollo del proyecto.

2.5.1 Historias de Usuario

Es una técnica utilizada para especificar los requisitos del *software*. Se trata de tarjetas de papel en las cuales se describe brevemente las características que el sistema debe poseer. Este artefacto es utilizado para realizar estimaciones de tiempo donde cada HU debe ser lo suficientemente comprensible para que los programadores puedan implementarlas con facilidad. Estas son descompuestas en tareas de programación para luego ser implementadas por programadores durante una iteración.

A continuación se exponen las HU Insertar servicio e Insertar área funcional:

Historia de Usuario	
Número: HU_17	Nombre Historia de Usuario: Insertar servicio.
Modificación de Historia de Usuario Número: 1	
Usuario: Maidelis Vega Calzado	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 3 días.

Riesgo en Desarrollo: Alto	Puntos Reales: 2 días.
<p>Descripción: La HU permite insertar un nuevo servicio en el sistema. Para insertarlo se debe seleccionar la opción “Administración” en el menú principal o en el menú superior, luego “Servicios”, posteriormente se selecciona “Nuevo Servicio”, se entran los datos y por último se selecciona el botón “Crear”.</p> <p>Para insertar un nuevo Servicio al sistema se deben insertar los siguientes datos:</p> <ul style="list-style-type: none"> - Nombre. - Descripción. - Lugar. - Fecha de Inicio. - Fecha fin. - Involucrados. - Propietario. - Cliente. - Urgencia. - Nombre del Acuerdo de Nivel de Servicio. - Áreas funcionales. - Departamento. - Impacto. - Precio. - Archivo. 	
<p>Observaciones:</p> <ol style="list-style-type: none"> 1- Todos los campos son obligatorios. 2- Se debe estar autenticado en el sistema. 3- Se debe poseer el rol de Administrador, Especialista o Técnico. 	

Tabla 2: HU Insertar Servicio.

Historia de Usuario	
Número: HU_21	Nombre Historia de Usuario: Insertar Área funcional.
Modificación de Historia de Usuario Número: 1	
Usuario: Maidelis Vega Calzado	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 3 días.
Riesgo en Desarrollo: Alto	Puntos Reales: 2 días.
<p>Descripción: La HU permite insertar una nueva área funcional en el sistema. Para insertarla se debe seleccionar la opción “Administración” en el menú principal o en el menú superior, luego “Áreas Funcionales de los Servicios”, posteriormente se selecciona “Nueva Área Funcional”, se entran los datos y por último se selecciona el botón “Crear”.</p> <p>Para insertar una nueva Área al sistema se deben insertar los siguientes datos:</p> <ul style="list-style-type: none"> - Nombre. - Descripción. 	

Observaciones:

- 1- Todos los campos son obligatorios.
- 2- Se debe estar autenticado en el sistema.
- 3- Se debe poseer el rol de Administrador, Especialista o Técnico.

Tabla 3: HU Insertar área funcional.

Las restantes HU aparecen reflejadas en el documento **Historias de usuario del subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio**, el cual se encuentra en el expediente de proyecto.

2.6 Fase de planificación

En la fase de planificación se establece la prioridad a cada HU creada y se realiza la estimación de esfuerzo necesaria para cada una de ellas. La planificación se puede realizar basándose en el tiempo o el alcance, al planificar por tiempo se determina cuántos puntos se puede completar, sin embargo al planificar según el alcance se obtiene el número de iteraciones necesarias para la implementación. Es necesario establecer cuántas HU se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de ellas.

2.6.1 Estimación de esfuerzo

La estimación de proyectos de *software* es un proceso de suma importancia debido a que permite hacer estimaciones razonables de recursos, duración, costo, etc. obteniéndose de esta manera una mayor visión del esfuerzo y tiempo necesario para concluir exitosamente una actividad o tarea determinada. Para tener una medida del progreso real del presente trabajo se realizó la estimación de esfuerzo por HU. Esta estimación fue establecida por el equipo de desarrollo utilizándose como medida el punto, el cual equivale a una semana ideal de programación. Se arrojaron los resultados que se muestran a continuación:

No.	Historias de Usuarios a implementar	Punto de estimación
1	Insertar usuario.	0.4
2	Modificar usuario.	0.4
3	Eliminar usuario.	0.4
4	Mostrar usuario.	0.3
5	Insertar grupo de usuarios.	0.4
6	Modificar grupo de usuarios.	0.4
7	Eliminar grupo de usuarios.	0.4
8	Mostrar grupo de usuarios.	0.3
9	Insertar rol de usuario.	0.4
10	Modificar rol de usuario.	0.4
11	Eliminar rol de usuario.	0.4
12	Mostrar rol de usuario.	0.3

13	Insertar acuerdo de nivel de servicio.	0.4
14	Modificar acuerdo de nivel de servicio.	0.4
15	Eliminar acuerdo de nivel de servicio.	0.4
16	Mostrar acuerdo de nivel de servicio.	0.3
17	Insertar servicio.	0.4
18	Modificar servicio.	0.4
19	Eliminar servicio.	0.3
20	Mostrar servicio.	0.3
21	Insertar área funcional.	0.4
22	Modificar área funcional.	0.4
23	Eliminar área funcional.	0.3
24	Mostrar área funcional.	0.3
25	Insertar acuerdo de nivel de operación.	0.4
26	Modificar acuerdo de nivel de operación.	0.4
27	Eliminar acuerdo de nivel de operación.	0.3
28	Mostrar acuerdos de nivel de operación.	0.3
29	Insertar proveedor.	0.4
30	Modificar proveedor.	0.4
31	Eliminar proveedor.	0.3
32	Mostrar proveedor.	0.3
33	Mostrar catálogo de servicios.	0.3
34	Insertar evaluación.	0.4
35	Modificar evaluación.	0.4
36	Eliminar evaluación.	0.3
37	Mostrar evaluación.	0.3
38	Insertar urgencia.	0.4
39	Modificar urgencia.	0.4
40	Eliminar urgencia.	0.3
41	Mostrar urgencia.	0.3
42	Insertar impacto.	0.4
43	Modificar impacto.	0.4
44	Eliminar impacto.	0.3
45	Mostrar impacto.	0.3
46	Insertar prioridad.	0.4
47	Modificar prioridad.	0.4
48	Eliminar prioridad.	0.3
49	Mostrar prioridad.	0.3
50	Insertar departamento.	0.4
51	Modificar departamento.	0.4
52	Eliminar departamento.	0.3
53	Mostrar departamento.	0.3
54	Asociar a un servicio un acuerdo de nivel de servicio.	0.1
55	Asociar un servicio a un acuerdo de nivel de operación.	0.1
56	Asociar a un servicio una o más áreas funcionales.	0.1
57	Autenticar usuario.	0.1
58	Realizar búsqueda simple.	0.1
59	Realizar búsqueda avanzada.	0.3

60	Realizar notificaciones por correos a usuarios.	0.3
61	Realizar notificaciones por correos a un grupo de usuarios.	0.3
62	Asociar una evaluación a proveedores.	0.1
63	Realizar reportes de los acuerdos de niveles de servicios y acuerdos de niveles de operación.	0.3
64	Adjuntar un documento a los acuerdos de niveles de servicios.	0.1
65	Adjuntar un documento a los servicios.	0.1
66	Mostrar últimos acuerdos de niveles de servicios creados.	0.1
67	Mostrar últimos acuerdos de niveles de operación creados.	0.1
68	Mostrar matriz de asignación de prioridad.	0.1
69	Mostrar información del sitio (manual de ayuda).	0.1
70	Mostrar reportes gráficos.	0.1

Tabla 4: Estimación de esfuerzo por HU.

2.6.2 Plan de duración de iteraciones

El Plan de Iteraciones es un documento que especifica las HU a implementar en cada fase de iteración. Las HU deben ser desarrolladas en un ciclo de iteración determinado y de acuerdo al orden de dichas iteraciones. Para el desarrollo del sistema propuesto se definieron 3 iteraciones, las cuales se describen a continuación:

Iteración	Descripción
1	Esta iteración tiene como objetivo darle cumplimiento a todas las HU que son de gran importancia para la solución propuesta, es decir con prioridad muy alta. Esta iteración recoge las principales funcionalidades del sistema, pues hacen alusión a la gestión de Acuerdos de Nivel de Servicios, Acuerdos de Nivel de Operación, Servicios, Áreas funcionales, Proveedores. Además también se gestionan los grupos y roles de usuarios.
2	Esta iteración está enfocada en implementar todas las HU con prioridad alta en el negocio. Se enmarca en la gestión de Urgencia, Impacto, Departamento y Prioridad.
3	Esta iteración está centrada en desarrollar todas las HU con prioridad media en el negocio. Está enmarcada en la realización de búsqueda simple y avanzada, notificaciones por correo a un usuario o grupo determinado, mostrar reportes gráficos, entre otras.

Tabla 5: Descripción de iteraciones.

Luego de describir todas las iteraciones necesarias para el desarrollo del sistema, es necesario crear el Plan de Duración de Iteraciones, el cual permite conocer las HU que serán implementadas en cada iteración, así como el tiempo real de duración de cada una de ellas.

Iteraciones	Historias de Usuarios a implementar	Duración (semanas)
Iteración 1	<ol style="list-style-type: none">1. Insertar usuario.2. Modificar usuario.3. Eliminar usuario.4. Mostrar usuario.5. Insertar grupo de usuarios.6. Modificar grupo de usuarios.7. Eliminar grupo de usuarios.8. Mostrar grupo de usuarios.9. Insertar rol de usuario.10. Modificar rol de usuario.11. Eliminar rol de usuario.12. Mostrar rol de usuario.13. Insertar acuerdo de nivel de servicio.14. Modificar acuerdo de nivel de servicio.15. Eliminar acuerdo de nivel de servicio.16. Mostrar acuerdo de nivel de servicio.17. Insertar servicio.18. Modificar servicio.19. Eliminar servicio.20. Mostrar servicio.21. Insertar área funcional.22. Modificar área funcional.23. Eliminar área funcional.24. Mostrar área funcional.25. Insertar acuerdo de nivel de operación.26. Modificar acuerdo de nivel de operación.27. Eliminar acuerdo de nivel de operación.28. Mostrar acuerdo de nivel de operación.29. Insertar proveedor.30. Modificar proveedor.31. Eliminar proveedor.32. Mostrar proveedor.33. Insertar catálogo de servicios.34. Modificar catálogo de servicios.35. Eliminar catálogo de servicios.36. Mostrar catálogo de servicios.	12 semanas

Iteración 2	<ol style="list-style-type: none"> 1. Insertar evaluación. 2. Modificar evaluación. 3. Eliminar evaluación. 4. Mostrar evaluación. 5. Insertar urgencia. 6. Modificar urgencia. 7. Eliminar urgencia. 8. Mostrar urgencia. 9. Insertar impacto. 10. Modificar impacto. 11. Eliminar impacto. 12. Mostrar impacto. 13. Insertar prioridad. 14. Modificar prioridad. 15. Eliminar prioridad. 16. Mostrar prioridad. 17. Insertar departamento. 18. Modificar departamento. 19. Eliminar departamento. 20. Mostrar departamento. 21. Asociar a un servicio un acuerdo de nivel de servicio. 22. Asociar un servicio a un acuerdo de nivel de operación. 23. Asociar a un servicio una o más áreas funcionales. 	7 semanas
Iteración 3	<ol style="list-style-type: none"> 1. Autenticar usuario. 2. Realizar búsqueda simple. 3. Realizar búsqueda avanzada. 4. Realizar notificaciones por correos a usuarios. 5. Realizar notificaciones por correos a un grupo de usuarios. 6. Asociar una evaluación a proveedores. 7. Realizar reportes de los acuerdos de niveles de servicios y acuerdos de niveles de operación. 8. Adjuntar un documento a los acuerdos de niveles de servicios. 9. Adjuntar un documento a los servicios. 10. Mostrar últimos acuerdos de niveles de servicios creados. 11. Mostrar últimos acuerdos de niveles de operación creados. 12. Mostrar matriz de asignación de prioridad. 13. Mostrar información del sitio (manual de ayuda). 14. Mostrar reportes gráficos 	3 semanas

Tabla 6: Plan de duración de iteraciones.

2.6.3 Plan de entrega

En el Plan de Entrega se define el orden y se agrupan las HU necesarias para conformar una entrega. Para la creación de este plan se debe tener en cuenta el grado de importancia de las HU para el cliente y el tiempo de desarrollo ideal. A continuación se muestra el plan de entrega del sistema propuesto, las HU se agruparon en tres entregas donde se tuvo en cuenta su nivel de prioridad para el negocio.

Producto a entregar	Final de la 1ra iteración 22 de febrero del 2014	Final de la 2da iteración 20 de abril del 2014	Final de la 3ra iteración 15 de mayo del 2014
GESERV	Versión 1.0	Versión 1.1	Versión 1.2

Tabla 7: Plan de Entrega.

2.7 Diseño del sistema

La metodología XP se enfoca en lograr diseños simples y sencillos, esto se debe en su totalidad, porque al tener un diseño fácilmente entendible se garantiza la implementación de la solución en menos tiempo y esfuerzo. También se recomienda no añadir funcionalidades extras, porque pocas veces son utilizadas, por tanto añadir estas funcionalidades implica exceso de tiempo y recursos.

2.7.1 Tarjetas CRC del sistema

Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. Una clase es cualquier persona, cosa, evento, concepto, pantalla o reporte. Las responsabilidades de una clase son las cosas que realizan, sus atributos y métodos, es decir define qué debe hacer. Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades, es decir todas las clases con las que interactúa. A continuación se muestran las Tarjetas CRC para la clase ANSController y ANS respectivamente.

Nombre de la clase: ANSController	
Responsabilidades	Clases relacionadas
Se encarga de generar los reportes de ANS en diferentes formatos. Además genera las funcionalidades: <ul style="list-style-type: none"> - List(integer max) - Create() - Save() - Show(long id) - Edit(long id) - Update(long id, long version) - Delete(long id) 	ANS.

Tabla 8: Tarjeta CRC de ANSController.

Nombre de la clase: ANS	
Responsabilidades	Clases relacionadas
Es la encargada de establecer los acuerdos necesarios con los	Impacto.

clientes y los proveedores para ofrecer los servicios requeridos, esta recoge mediante un lenguaje comprensible para el cliente todos los detalles de los servicios brindados.	Prioridad. Urgencia.
--	-------------------------

Tabla 9: Tarjeta CRC de ANS.

Las restantes Tarjetas CRC están reflejada en el documento **Tarjetas CRC del subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio**, el cual se encuentra en el expediente de proyecto.

2.7.2 Patrones de diseño utilizados

“Un patrón de diseño provee un esquema para refinar componentes de un sistema de software y la forma en que se relacionan entre sí. Describe una estructura, generalmente recurrente de comunicación de componentes que resuelve un problema de diseño general dentro de un contexto particular. Los patrones de diseño son patrones de granularidad media, ya que tienen menor nivel de abstracción que los patrones arquitectónicos pero tienden a ser independientes de un lenguaje de programación en particular o de un paradigma de programación” (Almeira and Pérez Cavenago 2007).

En el diseño del sistema se hará uso de los patrones Experto, Creador, Alta Cohesión, Bajo Acoplamiento y Controlador, los cuales son patrones GRASP (por sus siglas en español Patrones Generales de Software para Asignar Responsabilidades) y permiten la descripción de los principios fundamentales de la asignación de responsabilidades a objetos. A continuación se enuncian las principales características de cada uno de ellos:

Experto: Indica que la responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos), para obtener un diseño con mayor cohesión. Con la utilización de este patrón se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. El comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clases sencillas y más cohesivas que son más fáciles de comprender y mantener.

Creador: Establece el principio de la asignación de las responsabilidades relacionadas con la creación de objetos. La utilización de este patrón garantiza un bajo acoplamiento lo que significa menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización, además de mayor encapsulación. El uso del patrón Creador se evidencia en todas las clases de la capa controladora de la solución propuesta.

Alta Cohesión: 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. Mediante su uso se mejora la claridad y facilidad con que se entiende el diseño, simplificándose el mantenimiento y las mejoras en funcionalidades.

Bajo Acoplamiento: Su idea principal es tener las clases lo menos ligadas entre sí, lo cual garantiza que al producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases.

Controlador: Este patrón es el encargado de asignar la responsabilidad de controlar el flujo de eventos del sistema a clases específicas, facilitando la centralización de actividades. El patrón Controlador sugiere que la lógica de negocio debe estar separada de la capa de presentación para aumentar la reutilización de código y para tener mayor control.

2.7.3 Patrón de arquitectura

Cuando se desarrolla un *software* es necesaria la selección de un patrón arquitectónico. *“Estos patrones capturan existencia, experiencia comprobada en el desarrollo del software y ayudan a promover buenas prácticas de diseño. Se proponen como descripción de un problema particular y recurrente de diseño, que aparece en contextos de diseño específico y presenta un esquema genérico demostrado con éxito para su solución. El esquema de solución se especifica mediante la descripción de los componentes que la constituyen, sus responsabilidades y desarrollos”* (Almeira and Pérez Cavenago 2007).

Para la implementación del sistema se seleccionó como marco de trabajo Grails, el mismo define la utilización de patrón arquitectónico MVC. Este patrón separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes diferentes. Su principal propósito es aislar los cambios, desacoplar los datos, la lógica de negocio y la lógica de presentación, permitiendo la actualización y desarrollo independiente de cada uno de los citados componentes (Céspedes, 2007). En la figura 5 se muestran los niveles por lo que está formado este patrón arquitectónico.



Figura 6: Estructura del patrón MVC.

Modelo: Contiene la funcionalidad central y los datos.

Vista: Muestra la información del modelo al usuario, normalmente en HTML.

Controlador: Responde a las peticiones de los usuarios, controlando el flujo entre la vista y el modelo.

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo, independientemente de la representación visual.

2.8 Diseño de la base de datos

El modelado de datos es una de las actividades más importantes en la realización de una aplicación con base de datos, esto se debe en su totalidad, a que es el encargado de determinar su estructura, específicamente la manera en que se almacena, organizan y se manipulan todos los datos. Además responden a una serie de preguntas importantes en cualquier aplicación de procesamiento de datos.

La base de datos cuenta con 15 entidades persistentes, siendo la entidad Servicio la de mayor importancia, teniendo en cuenta que la mayoría de las restantes entidades tienen una relación directa con ella. Estas relaciones vienen dadas porque para la creación de un servicio es necesaria la asignación de un área funcional, proveedor, departamento, impacto, categoría, acuerdo de nivel de servicio y de operación. El cliente podrá acceder a la información correspondiente del sistema en dependencia de los roles y permisos establecidos.

El Modelo de Datos de la solución propuesta se encuentra reflejado en el [Anexo 5](#).

2.9 Conclusiones parciales

- La realización del modelo conceptual permitió una mayor comprensión del proceso de negocio en el Centro de Soporte UCI y la posterior identificación de clases importantes para la solución propuesta.
- A partir de los requisitos funcionales del sistema se describieron las HU, lo que proporcionó conocimiento del tiempo que conllevará la implementación de cada una de ellas.
- La estimación de esfuerzo posibilitó la identificación del tiempo necesario para el desarrollo de la solución propuesta.
- La realización de las Tarjetas CRC permitió la identificación de las clases, sus responsabilidades y todas las clases con las que trabaja en conjunto para llevar a cabo dicha responsabilidad, lo que permitió un mejor diseño del sistema.
- Finalmente, con la utilización de los patrones de diseño se logró el aporte de elementos reutilizables en el diseño de la solución propuesta, además de una búsqueda de soluciones a problemas comunes en el proceso de desarrollo de *software*.

IMPLEMENTACIÓN Y VALIDACIÓN DEL SUBSISTEMA DE GESTIÓN DE CATÁLOGOS Y GESTIÓN DE NIVEL DE SERVICIO



Cada paso durante el ciclo de desarrollo del *software* cumple un papel primordial para obtener un producto final con mayor calidad. En la fase de implementación, se procede a escribir el código necesario para obtener un producto final que cumpla con los requerimientos de software y responda al diseño del sistema. La validación mejora la calidad del producto desarrollado, asegura que satisfaga las necesidades del cliente y que cumpla con las especificaciones establecidas.

En el presente capítulo se identifican los estándares de codificación necesarios para la obtención de un código consistente. Se descomponen las HU en tareas de ingeniería y se analiza la calidad del Subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio, mediante la realización de pruebas unitarias, pruebas de funcionalidad, pruebas de rendimiento y pruebas de integración.

3.1 Implementación del sistema

Para lograr una exitosa implementación del sistema, XP define una docena de prácticas que propician un aumento en la productividad del software, dentro de estas se encuentran los estándares de codificación y la programación en parejas. La primera, se centra en conseguir que el código se encuentre en buen estado y que cualquier integrante del equipo de desarrollo pueda modificarlo, en caso necesario. La segunda, tiene como objetivo el trabajo en parejas para proporcionar un mecanismo de seguridad enormemente valioso.

3.1.1 Estándares de codificación

Los 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. El uso de estos estándares ayudan a mejorar el proceso de codificación haciéndolo más eficiente, además de servirle a los programadores como punto de referencia.

Para conseguir que el código se encuentre en buen estado y que cualquier persona del equipo pueda modificar cualquier parte de este, es imprescindible que el estilo de codificación sea consistente (Sánchez González 2004). Para la implementación del sistema se utilizaron los siguientes estándares de codificación:

Notación CamelCase: Es la nomenclatura por excelencia en el mundo Java, consiste en utilizar las mayúsculas como separadores de palabras. Si comienza con mayúscula, se denomina *UpperCamelCase*

Capítulo 3: Implementación y validación

y, si no, *lowerCamelCase*. En la solución propuesta se utilizó *lowerCamelCase* para nombrar los atributos de las clases. En la siguiente figura se evidencia el uso de este estándar:

```
class Servicios {
  ANS ans
  statichasMany = [categorias: Categorías]
  String nombre
  String descripción
  String lugar
}
```

Figura 7: Uso de la notación CamelCase.

Notación Snake-case: Más utilizada en lenguajes de *scripting*, esta notación hace uso de los guiones bajos para separar las palabras (generalmente en minúsculas). Snake-case fue usado para nombrar los atributos de las clases. La siguiente figura evidencia el uso de esta notación.

```
package aplicacion
class Proveedores {
  statichasMany = [servicios: Servicios]
  String tipo_adquisición
  String nombre_proveedor
  String descripción
  String nombre_contactos
  static ans
}
```

Figura 8: Uso de la notación Snake-case.

Notación Pascal Casing: Es un procedimiento de programación común en el lenguaje Java y .Net. La nomenclatura está compuesta por tantas palabras como sean necesarias. La primera letra de cada una de las palabras irá siempre en mayúsculas, obviando el uso de artículos. En la solución propuesta se hace uso de Pascal Casing para nombrar las clases que conforman el dominio.

```
class UsuarioController {
  static allowedMethods = [save: ""]
  def index() {redirect(action: "1")}
  def list(Integer max) {...}
}
```

Figura 9: Uso de la notación Pascal Casing.

3.1.2 Programación en parejas

Toda la producción de código será realizado en parejas de programadores, trabajando conjuntamente en un mismo ordenador para garantizar múltiples ventajas como mayor calidad del *software*, mejor diseño, transferencia de conocimientos de programación entre los miembros del equipo, las tasas de errores del producto final son más bajas, mayor satisfacción de los programadores, entre otras.

Este equipo debe poseer conocimientos similares de programación. Mientras uno de ellos se encarga de pensar la táctica que va a darle solución al problema, el otro se encarga de pensar las estrategias que permiten llevar dichas tácticas a su máximo exponente. Ambos roles son intercambiables.

3.1.3 Tareas de ingeniería

Durante la implementación del sistema, se van desarrollando las HU pertenecientes a una iteración determinada. Al principio de cada iteración se lleva a cabo una revisión del plan de iteraciones y se modifica si es necesario. Las HU son descompuestas en tareas de ingeniería, en las cuales se debe especificar la fecha de inicio, la fecha fin de la tarea, el programador responsable de cumplirla y se debe describir lo que se hará en la misma. Los parámetros específicos que se recogen en este documento son:

- **Número de la tarea:** Identificador de la tarea a realizar.
- **Número de HU:** Identifica la HU con la que se relaciona la tarea.
- **Nombre de la tarea:** Nombre de la tarea que se realiza.
- **Tipo de Tarea:** Precisa en que parte del ciclo de desarrollo se realiza la tarea.
- **Puntos estimados:** Tiempo estimado que debe durar el cumplimiento de la tarea.
- **Fecha de inicio:** Teniendo en cuenta el tiempo estimado, refleja la fecha dentro del cronograma en la que debe comenzar la realización de la tarea.
- **Fecha de fin:** Teniendo en cuenta el tiempo estimado, refleja la fecha dentro del cronograma en la que debe culminar la realización de la tarea.
- **Responsable:** Nombre de la persona a cargo de la realización de la tarea.
- **Descripción:** Describe el objetivo y las características de la tarea.

A continuación quedan detalladas las tareas de ingeniería realizadas para las HU Insertar servicio, Insertar evaluación y Asociar evaluación a un proveedor respectivamente.

Iteración 1

En esta iteración se le da cumplimiento a las HU con prioridad muy alta. La cual comprende gestionar usuarios, grupo de usuarios, roles de usuarios, acuerdos de niveles de servicios, acuerdos de niveles de operación, áreas funcionales, servicios y proveedores.

Tareas de Ingeniería	
Número de tarea: 1	Número de Historia de Usuario: 17
Nombre de la tarea: Implementar funcionalidad insertar servicio.	
Tipo de tarea: Desarrollo	Puntos estimados: 1

Capítulo 3: Implementación y validación

Fecha de inicio: 28/11/2013	Fecha fin: 28/11/2013
Programador responsable: Emil Pérez Piz.	
Descripción: Implementar la funcionalidad para que el administrador pueda insertar un nuevo servicio al sistema.	

Tabla 10: Tarea de ingeniería 1 para la HU Insertar servicio.

Tareas de Ingeniería	
Número de tarea: 2	Número de Historia de Usuario: 17
Nombre de la tarea: Crear interfaz de insertar servicio.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 29/11/2013	Fecha fin: 29/11/2013
Programador responsable: Emil Pérez Piz y Maidelis Vega Calzado.	
Descripción: Esta tarea permite la creación de una interfaz para la inserción de servicios. Estos datos serán insertados en un formulario.	

Tabla 11: Tarea de ingeniería 2 para la HU Insertar servicio.

Tareas de Ingeniería	
Número de tarea: 3	Número de Historia de Usuario: 17
Nombre de la tarea: Inserción de los datos de un servicio a la base de datos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 30/11/2013	Fecha fin: 30/11/2013
Programador responsable: Emil Pérez Piz y Maidelis Vega Calzado.	
Descripción: Esta tarea permite que luego de haber introducido todos los datos de un servicio de forma correcta en el formulario se almacenen estos datos en la base de datos.	

Tabla 12: Tarea de ingeniería 3 para la HU Insertar servicio.

Capítulo 3: Implementación y validación

Iteración 2

En esta iteración se le da cumplimiento a las HU con prioridad alta. La cual comprende gestionar evaluación, impacto, urgencia, prioridad, departamento y asociarle a un servicio un acuerdo de nivel de servicio, de operación y un área funcional.

Tareas de Ingeniería	
Número de tarea: 1	Número de Historia de Usuario: 34
Nombre de la tarea: Implementar funcionalidad insertar evaluación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 06/01/2014	Fecha fin: 06/01/2014
Programador responsable: Emil Pérez Piz	
Descripción: Implementar la funcionalidad para que el administrador pueda insertar una nueva evaluación al sistema.	

Tabla 13: Tarea de ingeniería 1 para la HU Insertar evaluación.

Tareas de Ingeniería	
Número de tarea: 2	Número de Historia de Usuario: 34
Nombre de la tarea: Crear interfaz de insertar evaluación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 07/01/2014	Fecha fin: 07/01/2014
Programador responsable: Emil Pérez Piz y Maidelis Vega Calzado.	
Descripción: Esta tarea permite la creación de una interfaz para la inserción de evaluaciones. Estos datos serán insertados en un formulario.	

Tabla 14: Tarea de ingeniería 2 para la HU Insertar evaluación.

Capítulo 3: Implementación y validación

Tareas de Ingeniería	
Número de tarea: 3	Número de Historia de Usuario: 34
Nombre de la tarea: Inserción de los datos de una evaluación a la base de datos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 08/01/2014	Fecha fin: 08/01/2014
Programador responsable: Emil Pérez Piz y Maidelis Vega Calzado.	
Descripción: Esta tarea permite que luego de haber introducido todos los datos de una evaluación de forma correcta en el formulario se almacenen estos datos en la base de datos.	

Tabla 15: Tarea de ingeniería 3 para la HU Insertar evaluación.

Iteración 3

En esta iteración se le da cumplimiento a las HU con prioridad media. La cual comprende autenticar usuario, realizar búsqueda simple y avanzada, realizar notificaciones por correo, asociar una evaluación a un proveedor, realizar reportes gráficos, etc.

Tareas de Ingeniería	
Número de tarea: 1	Número de Historia de Usuario: 62
Nombre de la tarea: Asociar evaluación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 07/04/2014	Fecha fin: 07/04/2014
Programador responsable: Emil Pérez Piz	
Descripción: Implementar la funcionalidad insertar proveedor para que el administrador pueda insertar un nuevo proveedor y pueda asociarle una evaluación creada con anterioridad.	

Tabla 16: Tarea de ingeniería 1 para la HU Asociar una evaluación.

Las restantes Tareas de ingeniería están reflejada en el documento **Tareas de ingeniería del subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio**, el cual se encuentra en el expediente de proyecto, el cual se encuentra en el expediente de proyecto.

3.2 Pruebas de software

En el proceso de desarrollo de *software* es necesaria la realización de una serie de actividades encaminadas a encontrar errores en el mismo. Para garantizar la calidad de un sistema, se deben corregir estos errores, para esto se utilizan las pruebas. El uso de pruebas constituye un pilar fundamental de la metodología XP, permitiendo el aumento de la calidad del sistema, reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces, cuando esto sucede se puede decir que la prueba ha tenido éxito. Si el funcionamiento del *software* parece ser correcto y los errores encontrados son fáciles de corregir, se puede concluir con que la calidad y la fiabilidad del mismo son aceptables, o que las pruebas son inadecuadas para descubrir errores serios.

3.2.1 Pruebas unitarias

Las pruebas unitarias tienen como objetivo verificar que un método se comporta como debería, sin tener en cuenta su entorno, probando cada método de las clases implementadas por separado. Mediante estas pruebas se define como debe comportarse el sistema, de manera que ese comportamiento se mantenga con el tiempo. Además, al encontrarse un fallo en el *software* se construirá un caso de prueba que lo reproduzca y asegure que una vez resuelto este, no vuelva a aparecer.

Estas pruebas fueron realizadas a la solución propuesta mediante el uso del potente sistema de pruebas con el que cuenta Grails, aprovechando el soporte nativo en Groovy para distintas formas de Mocks¹⁵ y Stubs¹⁶. Se seleccionaron tres de las entidades más importantes del subsistema, ellas fueron RolController, Usuario y Categorías. En las siguientes figuras se muestran los resultados obtenidos en la realización de las pruebas unitarias a las clases anteriormente mencionadas.

¹⁵ Idéntico a un Stub, pero con la diferencia de que se añaden condiciones para comprobar la correcta utilización de la clase.

¹⁶ Permite que la prueba defina respuestas a llamadas de método (valores de retorno, excepciones) para simular el objeto dependiente.



Figura 10: Prueba unitaria aplicada a la clase RollController.

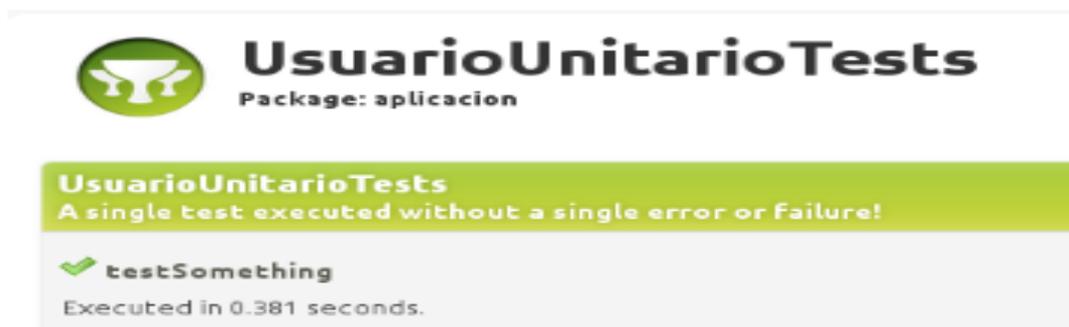


Figura 11: Prueba unitaria aplicada a la clase Usuario.

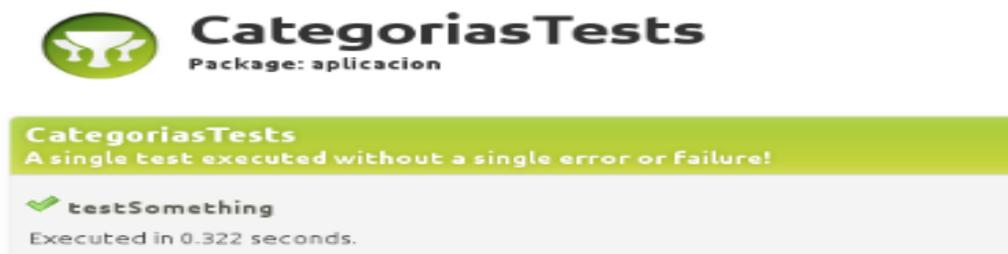


Figura 12: Prueba unitaria aplicada a la clase Categorías.

3.2.2 Pruebas de funcionalidad

Las pruebas de funcionalidad tienen como objetivo verificar los requisitos del sistema, los cuales son la principal fuente de información a la hora de construir estas pruebas. Además, se debe demostrar que las funciones del *software* son operativas, que la entrada se acepta de forma adecuada y que se produce una

Capítulo 3: Implementación y validación

salida correcta. Con el empleo de este tipo de prueba se pretende encontrar errores en la interfaz, en estructuras de datos o en accesos a bases de datos externas, de rendimiento, de inicialización y de terminación. Las pruebas de funcionalidad son realizadas a cada HU y es válido agregar que esta no se puede considerar terminada hasta que pase correctamente todas las pruebas.

Para cada uno de los casos de pruebas realizados al sistema se obtuvo una evaluación acorde a su resultado. La evaluación se describe a continuación:

- **Satisfactoria:** Cuando el resultado de la prueba realizada es el esperado.
- **Insatisfactoria:** Cuando el resultado de la prueba realizada muestra como resultados elementos erróneos, cuando se genera error de codificación o cuando el resultado no es completamente el esperado.

A continuación se ejemplifican 2 casos de pruebas realizados a las HU Insertar servicio e Insertar área funcional respectivamente:

Caso de prueba de aceptación	
Código: HU17_P1	HU: 17
Nombre: Insertar un servicio	
Descripción: Esta prueba tiene como objetivo evaluar el correcto funcionamiento de la opción insertar servicio.	
Condiciones de ejecución: Se debe poseer el rol de Técnico, Especialista o Administrador y se debe estar autenticado en el sistema.	
Procedimiento: <ul style="list-style-type: none">- El usuario selecciona la opción "Administración" en el menú principal.- El sistema debe mostrar una interfaz con todas las opciones administrativas.- Luego selecciona la opción "Servicios".- El sistema muestra un listado con todos los servicios creados anteriormente.- Luego selecciona la opción "Nuevo Servicio".- El sistema muestra un formulario con los siguientes datos: Nombre, Descripción, Lugar, Fecha de Inicio, Fecha fin, Involucrados, Propietario, Cliente, Urgencia, Nombre del Acuerdo de Nivel de Servicio, Áreas funcionales, Departamento, Impacto, Precio y Archivo.- El usuario inserta todos los datos de forma correcta en el formulario y luego da clic en el botón "Crear".- El sistema muestra los datos del nuevo servicio creado.	
Resultado esperado: Se inserta satisfactoriamente un nuevo servicio al sistema.	

Capítulo 3: Implementación y validación

Evaluación de la prueba: satisfactoria

Tabla 17: Caso de prueba 1 para la HU Insertar servicio.

Caso de prueba de aceptación	
Código: HU17_P2	HU: 17
Nombre: Insertar un servicio	
Descripción: Esta prueba tiene como objetivo evaluar el correcto funcionamiento de la opción insertar servicio.	
Condiciones de ejecución: Se debe poseer el rol de Técnico, Especialista o Administrador y se debe estar autenticado en el sistema.	
Procedimiento: <ul style="list-style-type: none">- El usuario selecciona la opción "Administración" en el menú principal.- El sistema debe mostrar una interfaz con todas las opciones administrativas.- Luego selecciona la opción "Servicios".- El sistema muestra un listado con todos los servicios creados anteriormente.- Luego selecciona la opción "Nuevo Servicio".- El sistema muestra un formulario con los siguientes datos: Nombre, Descripción, Lugar, Fecha de Inicio, Fecha fin, Involucrados, Propietario, Cliente, Urgencia, Nombre del Acuerdo de Nivel de Servicio, Áreas funcionales, Departamento, Impacto, Precio y Archivo.- El usuario no inserta todos los datos de forma correcta en el formulario y luego da clic en el botón "Crear".- No se inserta el nuevo servicio.	
Resultado esperado: El sistema muestra nuevamente la interfaz de insertar servicio, para que completen los campos que no han sido llenados o para que los inserte correctamente.	
Evaluación de la prueba: satisfactoria	

Tabla 18: Caso de prueba 2 para la HU Insertar servicio.

Caso de prueba de aceptación	
Código: HU21_P1	HU: 21
Nombre: Insertar un área funcional en el sistema.	
Descripción: Esta prueba tiene como objetivo evaluar el correcto funcionamiento de la opción insertar área	

Capítulo 3: Implementación y validación

funcional.
Condiciones de ejecución: Se debe poseer el rol de Técnico, Especialista o Administrador y se debe estar autenticado en el sistema.
Procedimiento: <ul style="list-style-type: none">- El usuario selecciona la opción “Administración” en el menú principal.- El sistema debe mostrar una interfaz con todas las opciones administrativas.- Luego selecciona la opción “Áreas funcionales de los servicios”.- El sistema muestra un listado con todas las áreas funcionales creadas anteriormente.- Luego selecciona la opción “Nuevo Áreas funcionales”.- El sistema muestra un formulario con los siguientes datos: Nombre y Descripción.- El usuario inserta todos los datos de forma correcta en el formulario y luego da clic en el botón “Crear”.- El sistema muestra los datos de la nueva área creada.
Resultado esperado: Se inserta satisfactoriamente una nueva área al sistema.
Evaluación de la prueba: satisfactoria

Tabla 19: Caso de prueba 1 para la HU Insertar área funcional.

Caso de prueba de aceptación	
Código: HU21_P2	HU: 21
Nombre: Insertar un área funcional al sistema.	
Descripción: Esta prueba tiene como objetivo evaluar el correcto funcionamiento de la opción insertar área funcional.	
Condiciones de ejecución: Se debe poseer el rol de Técnico, Especialista o Administrador y se debe estar autenticado en el sistema.	
Procedimiento: <ul style="list-style-type: none">- El usuario selecciona la opción “Administración” en el menú principal.- El sistema debe mostrar una interfaz con todas las opciones administrativas.- Luego selecciona la opción “Áreas funcionales de los servicios”.- El sistema muestra un listado con todas las áreas funcionales creadas anteriormente.- Luego selecciona la opción “Nueva Área funcional de los servicios”.- El sistema muestra un formulario con los siguientes datos: Nombre y Descripción.- El usuario no inserta todos los datos de forma correcta en el formulario y luego da clic en el botón “Crear”.	

- No se inserta la nueva área.
Resultado esperado: El sistema muestra nuevamente la interfaz de insertar áreas funcionales, para que completen los campos que no han sido llenados o para que los inserte correctamente.
Evaluación de la prueba: satisfactoria

Tabla 20: Caso de prueba 2 para la HU Insertar área funcional.

Los restantes Casos de pruebas se encuentran en el documento **Casos de prueba basados en HU del subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio**, el cual se encuentra en el expediente de proyecto.

Análisis de resultados

Se realizaron 133 casos de pruebas para validar que el sistema funcionara correctamente, mostrando las salidas correspondientes a cada escenario. De estas pruebas realizadas, en una primera iteración fueron clasificadas 95 pruebas de satisfactorias. En una segunda iteración, de un total de 38 pruebas realizadas fueron evaluadas de satisfactorias 24 de ellas. En una tercera y última iteración, de un total de 14 pruebas realizadas resultaron todas satisfactorias, constituyendo un 100% de pruebas funcionales exitosas. La siguiente figura muestra la cantidad de pruebas satisfactorias y no satisfactorias identificadas en cada iteración.

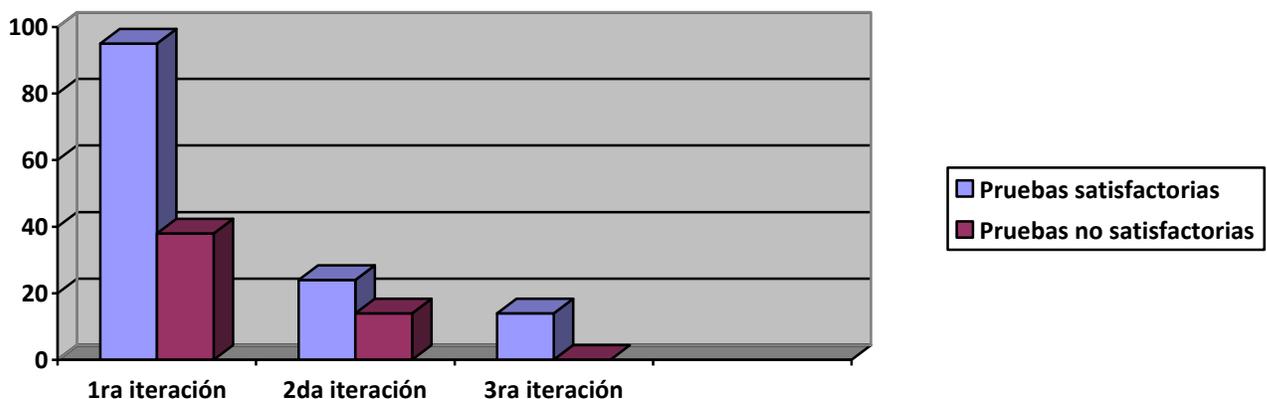


Figura 13: Resultado de las pruebas funcionales.

3.2.3 Pruebas de rendimiento

Las pruebas de rendimiento se utilizan para probar la aplicación bajo condiciones normales y de pico, lo que permite verificar que se cumplirán los objetivos deseados. En estas pruebas se fuerza a la aplicación

Capítulo 3: Implementación y validación

a funcionar más allá de las condiciones para las que fue diseñada (Lopez 2004). Además se pueden calcular ciertas características de la aplicación, entre ellas punto de ruptura del servidor, problemas de sincronización, pérdida de datos por congestión de la red, número máximo de usuarios concurrentes, uso de los recursos, tiempo de respuesta y peticiones atendidas por segundo.

Estas pruebas fueron realizadas a la aplicación con la herramienta de carga JMeter para 300 peticiones concurrentes, la utilización de esta herramienta permite la simulación de una amplia gama de recursos de *software* en tiempo de ejecución real, a partir de solicitudes sencillas hasta otras de mayor envergadura. Para la ejecución de la prueba fue usada una computadora servidor con un microprocesador Intel Pentium Inside (frecuencia: 2.2GHz), memoria RAM de 2GB y disco duro de 500GB. Como cliente fue usada una computadora con microprocesador Intel Dual Core (frecuencia: 2.0 GHz), memoria RAM de 1GB y disco duro de 180GB conectadas a través de tarjetas de red Fast Ethernet con velocidad de 100Mbps.

En la siguiente figura se muestran los resultados obtenidos para 300 peticiones concurrentes, la tabla completa se encuentra reflejada en el [Anexo 6](#).

Etiqueta	# Muestras	Media	Mediana	Linea de 9...	Mín	Máx	% Error	Rendimiento	Kb/sec
Total	10536	546	16	606	2	145837	0,00%	5,9/sec	927,4

Figura 14: Resultados obtenidos para 300 peticiones concurrentes.

Los resultados obtenidos en las pruebas de rendimiento se consideran satisfactorios, se generó un total de 927,4 kb/segundo de transferencia de datos para 300 usuarios concurrentes, lo que incurrió en un rendimiento de 5,9 segundos, menor que el especificado en los requerimientos de *software*. Se demuestra que la propuesta de solución es estable, ya que se mantuvo prestando servicios sin incurrir en fallos.

3.2.4 Pruebas de integración

Las pruebas de integración definen cómo debe comportarse el sistema de manera que ese comportamiento se mantenga con el tiempo. Esta prueba tiene como objetivo verificar el funcionamiento de los componentes de la aplicación mediante la combinación de los módulos individuales del software, para que sean probados como un grupo. Para la realización de las pruebas se crearon las baterías de pruebas correspondientes a las clases UsuariosController, ANSController y UrgenciasController. Se analizó el comportamiento de las acciones listar, mostrar, actualizar y eliminar correspondientes a cada una de ellas, respondiendo satisfactoriamente.

3.3 Conclusiones parciales

- La utilización de los estándares de codificación permitió la obtención de un código entendible, con mayor calidad y menor número de errores, lo que posibilita su reutilización y posterior mantenimiento.
- La práctica programación en parejas posibilitó mayor satisfacción de los programadores, el trabajo en equipo y la implementación del código en menos tiempo.
- La descomposición de las HU en tareas de ingeniería proporcionó menor tiempo y mayor destreza en la implementación de las mismas.
- Finalmente, con la realización de las pruebas de software se corrigieron los errores identificados, lo que aseguró la calidad y fiabilidad de la solución propuesta. Por consiguiente, se corroboró que la misma cumple satisfactoriamente con las funcionalidades identificadas durante el proceso de desarrollo del *software*.

Conclusiones

Con el propósito de darle cumplimiento al objetivo general planteado se han llevado a cabo satisfactoriamente cada una de las tareas de investigación definidas para el presente trabajo:

- Con el estudio de los procesos de Gestión de Catálogos y Gestión de Nivel de Servicio, se definió el marco teórico conceptual de la investigación, lo que permitió identificar la situación problemática y las bases para analizar y diseñar los procesos antes mencionados.
- El estudio del estado del arte permitió obtener las principales características y funcionalidades de los sistemas de Gestión de Catálogos y Gestión de Nivel de Servicio analizados. Se demostró la necesidad de desarrollar un Subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio para contribuir al correcto funcionamiento de estos procesos en el Centro de Soporte UCI.
- El análisis de las herramientas y tecnologías utilizadas en el proceso de desarrollo permitió describir sus principales características y definir las potencialidades que resultan ventajosas para la propuesta de solución, logrando una mejor comprensión y mayor familiarización de estas.
- La realización del análisis y diseño de la solución constituyó la base fundamental para las actividades de implementación y permitió la descomposición de las mismas en partes más manejables a desarrollar.
- Con la implementación de los componentes del modelo se obtuvo como resultado un Subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio, el cual contribuye al correcto funcionamiento de estos procesos en el Centro de Soporte UCI.
- El diseño y ejecución de los casos de prueba de las funcionalidades implementadas permitió la identificación y posterior corrección de las no conformidades. Posibilitando asegurar la calidad, eficiencia y correcto funcionamiento de los requisitos de *software* de la solución propuesta.

Recomendaciones

Con vista a mejorar el presente trabajo se recomienda:

- Integrar el subsistema de Gestión de Catálogos y Gestión de Nivel de Servicio con el resto de los subsistemas, que darán como resultado el software que se utilizará en el Centro de Soporte UCI, para brindar los servicios de soporte técnico a los productos informáticos desarrollados en la Universidad.

Bibliografía referenciada

1. ALMEIRA, A.S. and PÉREZ CAVENAGO, V. 2007. *Arquitectura de Software : Estilos y Patrones*. S.I.: Universidad Nacional de la Patagonia San Juan Bosco.
2. Apache Software Foundation. (2013). The Apache Software Foundation. Retrieved April 20, 2014, from <http://jmeter.apache.org/>
3. APACHE SOFTWARE FOUNDATION 2014. Apache Tomcat. [en línea]. [Consulta: 3 December 2013]. Disponible en: <http://tomcat.apache.org/>.
4. ÁREA DE AUDITORÍA Y CONTROL 2007. COBIT : Modelo para auditoría y control de sistemas de información. 54. May 2007. pp. 4.
5. AXIOS SYSTEMS 2011. ¿Habla ITIL? Assyst. S.I.:
6. BECK, K. 2004. *Extreme Programming Explained: Embrace Change, 2nd Edition*. 2nd. S.I.: Addison-Wesley; 2nd edition. ISBN 978-0321278654.
7. BH CONSULTING, 2005. *ITIL & BS 15000*. 2005. S.I.: s.n.
8. BMC SOFTWARE 2005. Remedy ITSM. [en línea]. [Consulta: 2 December 2013]. Disponible en: <http://www.bmc.com/products/remedy-itsm/it-service-management-suite.html>.
9. BOLOGNA, J. and WASH 1997. *The Accountant's Handbook of Information Technology*. 1. S.I.: Wiley; 1 edition (February 12, 1997). ISBN 978-0471304739.
10. BRITO, N. 2009a. *Manual de desarrollo web con Grails*. S.I.: s.n. ISBN 978-84-613-2651.
11. BRITO, N. 2009b. *Manual de desarrollo web con Grails*. ediciones . S.I.: s.n. ISBN 978-84-613-2651.
12. CALAFAT MESQUIDA, A.L. 2012. *Un Modelo para Facilitar la Integración de Estándares de Gestión de TI en Entornos Maduros*. S.I.: Universitat de les Illes Balears.
13. CARRILLO VERDÚN, J. and RUBIO CASALLAS, A.P. 2011. Modelo de Procesos Integrado de Gobernanza y Gestión de TI . Vol. 8, no. 1698-2029, pp. 17.
14. CÉSPEDES, E.S. 2007. *Batería para la realización de pruebas neuropsicológicas* . S.I.: Universidad de las Ciencias Informáticas.
15. COELLO, H. 2008. ITIL, COBIT, CMMI, PMBOK: Como integrar y adoptar los estándares para un buen Gobierno de TI. 8 de diciembre del 2008 [en línea]. [Consulta: 29 November 2013]. Disponible en: <http://helkyncoello.wordpress.com/2008/12/08/itil-cobit-cmmi-pmbok-como-integrar-y-adoptar-los-estandares-para-un-buen-gobierno-de-ti/>.
16. CORONA, M. 2011. ITIL edición 2011. S.I.:

17. DEFINICIÓN ABC 2007. Definiciones ABC. [en línea]. [Consulta: 28 January 2014]. Disponible en: <http://www.definicionabc.com/economia/servicio.php>.
18. FERNÁNDEZ ESCRIBANO, G. 2002. Introducción a Extreme Programming. S.l.: s.n. pp. 14.
19. HANNA, A. and RANCE, S., 2011. *glosario y abreviaturas de ITIL Español (Latinoamericano)*. 2011. S.l.: s.n.
20. HP SOFTWARE 2011. Introducción a HP Service Manager. 29 de agosto del 2011 [en línea]. [Consulta: 23 January 2014]. Disponible en: <http://www.stanum.es/noticias/118-introduccion-a-hp-service-manager.html>.
21. IBM ESPAÑA S.A 2011. Centro de servicios integrados y catálogo de servicios para una IT de “un toque”. [en línea]. Disponible en: <http://www-03.ibm.com/software/products/es/servicerequestmanager/>.
22. IT GOVERNANCE INSTITUTE 2006. *COBIT 4.0*. S.l.: s.n. ISBN 1-933284-37-4.
23. IT GOVERNANCE INSTITUTE, 2008a. *Alineando Cobit 4.1, Itil v3 e ISO/IEC 27002 en beneficio del negocio*. 2008. Austria: s.n.
24. IT GOVERNANCE INSTITUTE, 2008b. *Alineando Cobit 4.1, Itil v3 e ISO/IEC 27002 en beneficio del negocio*. 2008. Estados Unidos: s.n.
25. JARAMILLO GONZÁLEZ, M.I., TREJO RAMÍREZ, R.A. and BARRANCO GARCÍA, I., 2010. *Utilización de Estándares ITIL para lograr el Nivel 3 de CMMI en una Organización*. 2010. México: s.n.
26. JUDD, C.M., NUSAIRAT, J.F. and SHINGLER, J. 2008. *Beginning Groovy and Grails*. New York: s.n. ISBN 978-1-4302-1045-0.
27. LONG, J.O. 2011. *ITIL @ 2011 At a Glance*. New York: s.n. ISBN 978-1-4614-3897-7.
28. LOPEZ, D. 2004. Introducción a las pruebas de carga y stress de aplicaciones Web con Application Center Test. *dotNetManía*. Vol. 9, no. 1698-5451, pp. 58.
29. MARTÍNEZ, R. 2009. PostgreSQL-es. 2 de octubre del 2010 [en línea]. Disponible en: <http://www.postgresql.org.es/>.
30. OSIATIS S.A 2007a. ITIL®v3-Gestión de Servicios TI. *versión 3.0* [en línea]. [Consulta: 29 November 2013]. Disponible en: http://itilv3.osiatis.es/ciclo_vida_servicios_TI.php.
31. OSIATIS S.A 2007b. ITIL®v3-Gestión de Servicios TI. *versión 3.0* [en línea]. [Consulta: 29 November 2013]. Disponible en: http://itilv3.osiatis.es/disenio_servicios_TI/gestion_catalogo_servicios.php.
32. OSIATIS S.A 2013. ITIL®-Gestión de Servicios TI. *versión 2.0* [en línea]. [Consulta: 29 November 2013]. Disponible en: http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/gestion_de_niveles_de_servicio/introduccion_

- objetivos_gestion_de_niveles_de_servicio/introduccion_objetivos_gestion_de_niveles_de_servicio.php.
33. RAMÍREZ, Y. 2013. *Sistema de Gestión de Mapeo y Evaluación de Competencias del Sistema Automatizado de Monitoreo y Control de los Servicios para el Centro de Soporte de la Universidad de las Ciencias Informáticas*. S.l.: Universidad de las Ciencias Informáticas.
 34. REAL ACADEMIA ESPAÑOLA 2001. Diccionario de la Real Academia. 22 [en línea]. [Consulta: 27 November 2013]. Disponible en: <http://rae.es/recursos/diccionarios/drae>.
 35. REMEDIOS CRUZ, D. del P. 2012. *Guía de trabajo para las prácticas Diseño incremental y Pruebas continuas de la metodología XP para el diseño de las aplicaciones compuestas*. S.l.: Universidad de las Ciencias Informáticas.
 36. SALLÉ, M., 2004. *IT Service Management and IT Governance: Review, Comparative Analysis and their Impact on Utility Computing*. 2004. S.l.: s.n.
 37. SÁNCHEZ GONZÁLEZ, C. 2004. *ONess : un proyecto open source para el negocio textil mayorista desarrollado con tecnologías open source innovadoras* . S.l.: Universidade da Coruña.
 38. SOMMERVILLE, I. 2006. Software Requirements. En: 8th EDITION (ed.), *Software Engineering*. S.l.: s.n. pp. 25. ISBN 7-111-19770-4.
 39. TURBITT, K., 2006. *ISO 20000: ¿qué deben hacer las organizaciones?* 2006. S.l.: s.n.
 40. VISUAL PARADIGM 2010. Visual Paadigm. [en línea]. [Consulta: 20 January 2014]. Disponible en: <http://www.visual-paradigm.com>.
 41. VV.AA 2006. *UNE 50-113-91 2 DOCUMENTACION E INFORMACION*. 2006. Madrid, España: s.n. ISBN 2910011565628.

Bibliografía consultada

1. ALMEIRA, A.S. and PÉREZ CAVENAGO, V. 2007. *Arquitectura de Software : Estilos y Patrones*. S.I.: Universidad Nacional de la Patagonia San Juan Bosco.
2. Apache Software Foundation. (2013). The Apache Software Foundation. Retrieved April 20, 2014, from <http://jmeter.apache.org/>
3. APACHE SOFTWARE FOUNDATION 2014. Apache Tomcat. [en línea]. [Consulta: 3 December 2013]. Disponible en: <http://tomcat.apache.org/>.
4. ÁREA DE AUDITORÍA Y CONTROL 2007. COBIT : Modelo para auditoría y control de sistemas de información. 54. May 2007. pp. 4.
5. AXIOS SYSTEMS 2011. ¿Habla ITIL? Assyst. S.I.:
6. BECK, K. 2004. *Extreme Programming Explained: Embrace Change, 2nd Edition*. 2nd. S.I.: Addison-Wesley; 2nd edition. ISBN 978-0321278654.
7. BH CONSULTING, 2005. *ITIL & BS 15000*. 2005. S.I.: s.n.
8. BMC SOFTWARE 2005. Remedy ITSM. [en línea]. [Consulta: 2 December 2013]. Disponible en: <http://www.bmc.com/products/remedy-itsm/it-service-management-suite.html>.
9. BOLOGNA, J. and WASH 1997. *The Accountant's Handbook of Information Technology*. 1. S.I.: Wiley; 1 edition (February 12, 1997). ISBN 978-0471304739.
10. BRITO, N. 2009a. *Manual de desarrollo web con Grails*. S.I.: s.n. ISBN 978-84-613-2651.
11. BRITO, N. 2009b. *Manual de desarrollo web con Grails*. ediciones . S.I.: s.n. ISBN 978-84-613-2651.
12. CALAFAT MESQUIDA, A.L. 2012. *Un Modelo para Facilitar la Integración de Estándares de Gestión de TI en Entornos Maduros*. S.I.: Universitat de les Illes Balears.
13. CARRILLO VERDÚN, J. and RUBIO CASALLAS, A.P. 2011. Modelo de Procesos Integrado de Gobernanza y Gestión de TI . Vol. 8, no. 1698-2029, pp. 17.
14. CÉSPEDES, E.S. 2007. *Batería para la realización de pruebas neuropsicológicas* . S.I.: Universidad de las Ciencias Informáticas.
15. COELLO, H. 2008. ITIL, COBIT, CMMI, PMBOK: Como integrar y adoptar los estándares para un buen Gobierno de TI. 8 de diciembre del 2008 [en línea]. [Consulta: 29 November 2013]. Disponible en: <http://helkyncoello.wordpress.com/2008/12/08/itil-cobit-cmmi-pmbok-como-integrar-y-adoptar-los-estandares-para-un-buen-gobierno-de-ti/>.
16. Corbea, M. R., & Pérez, M. O. (2007). *La metodología XP aplicable al desarrollo del software educativo en Cuba*. Universidad de las Ciencias Informáticas.

17. CORONA, M. 2011. ITIL edición 2011. S.I.:
18. DEFINICIÓN ABC 2007. Definiciones ABC. [en línea]. [Consulta: 28 January 2014]. Disponible en: <http://www.definicionabc.com/economia/servicio.php>.
19. Estrada, A. F., Michaelucicu, M. G., & Napal, I. (2007). Enseñanza de las pruebas de software desde la producción. La Habana.
20. FERNÁNDEZ ESCRIBANO, G. 2002. Introducción a Extreme Programming. S.I.: s.n. pp. 14.
21. HANNA, A. and RANCE, S., 2011. *glosario y abreviaturas de ITIL Español (Latinoamericano)*. 2011. S.I.: s.n.
22. HP SOFTWARE 2011. Introducción a HP Service Manager. 29 de agosto del 2011 [en línea]. [Consulta: 23 January 2014]. Disponible en: <http://www.stanum.es/noticias/118-introduccion-a-hp-service-manager.html>.
23. IBM ESPAÑA S.A 2011. Centro de servicios integrados y catálogo de servicios para una IT de “un toque”. [en línea]. Disponible en: <http://www-03.ibm.com/software/products/es/servicerequestmanager/>.
24. IT GOVERNANCE INSTITUTE 2006. *COBIT 4.0*. S.I.: s.n. ISBN 1-933284-37-4.
25. IT GOVERNANCE INSTITUTE, 2008a. *Alineando Cobit 4.1, Itil v3 e ISO/IEC 27002 en beneficio del negocio*. 2008. Austria: s.n.
26. IT GOVERNANCE INSTITUTE, 2008b. *Alineando Cobit 4.1, Itil v3 e ISO/IEC 27002 en beneficio del negocio*. 2008. Estados Unidos: s.n.
27. JARAMILLO GONZÁLEZ, M.I., TREJO RAMÍREZ, R.A. and BARRANCO GARCÍA, I., 2010. *Utilización de Estándares ITIL para lograr el Nivel 3 de CMMI en una Organización*. 2010. México: s.n.
28. JUDD, C.M., NUSAIRAT, J.F. and SHINGLER, J. 2008. *Beginning Groovy and Grails*. New York: s.n. ISBN 978-1-4302-1045-0.
29. Larman, C. (2003). Modelo de dominio. In P. Hall (Ed.), *UML y Patrones* (2nd edition., pp. 1–23).
30. LONG, J.O. 2011. *ITIL @2011 At a Glance*. New York: s.n. ISBN 978-1-4614-3897-7.
31. LOPEZ, D. 2004. Introducción a las pruebas de carga y stress de aplicaciones Web con Application Center Test. *dotNetManía*. Vol. 9, no. 1698-5451, pp. 58.
32. MARTÍNEZ, R. 2009. PostgreSQL-es. 2 de octubre del 2010 [en línea]. Disponible en: <http://www.postgresql.org.es/>.
33. OSIATIS S.A 2007a. ITIL@v3-Gestión de Servicios TI. *versión 3.0* [en línea]. [Consulta: 29 November 2013]. Disponible en: http://itilv3.osiatis.es/ciclo_vida_servicios_TI.php.

34. OSIATIS S.A 2007b. ITIL®v3-Gestión de Servicios TI. *versión 3.0* [en línea]. [Consulta: 29 November 2013]. Disponible en:
http://itilv3.osiatis.es/disenio_servicios_TI/gestion_catalogo_servicios.php.
35. OSIATIS S.A 2013. ITIL®-Gestión de Servicios TI. *versión 2.0* [en línea]. [Consulta: 29 November 2013]. Disponible en:
http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/gestion_de_niveles_de_servicio/introduccion_objetivos_gestion_de_niveles_de_servicio/introduccion_objetivos_gestion_de_niveles_de_servicio.php.
36. RAMÍREZ, Y. 2013. *Sistema de Gestión de Mapeo y Evaluación de Competencias del Sistema Automatizado de Monitoreo y Control de los Servicios para el Centro de Soporte de la Universidad de las Ciencias Informáticas*. S.I.: Universidad de las Ciencias Informáticas.
37. REAL ACADEMIA ESPAÑOLA 2001. Diccionario de la Real Academia. 22 [en línea]. [Consulta: 27 November 2013]. Disponible en: <http://rae.es/recursos/diccionarios/drae>.
38. REMEDIOS CRUZ, D. del P. 2012. *Guía de trabajo para las prácticas Diseño incremental y Pruebas continuas de la metodología XP para el diseño de las aplicaciones compuestas*. S.I.: Universidad de las Ciencias Informáticas.
39. SALLÉ, M., 2004. *IT Service Management and IT Governance: Review, Comparative Analysis and their Impact on Utility Computing*. 2004. S.I.: s.n.
40. SÁNCHEZ GONZÁLEZ, C. 2004. *ONess : un proyecto open source para el negocio textil mayorista desarrollado con tecnologías open source innovadoras* . S.I.: Universidade da Coruña.
41. SOMMERVILLE, I. 2006. Software Requirements. En: 8th EDITION (ed.), *Software Engineering*. S.I.: s.n. pp. 25. ISBN 7-111-19770-4.
42. Soporte Remoto de México, S. A. de C. V. (2008). ¿Qué es ITIL? Ventajas y Desventajas. Retrieved from http://www.sopoteremoto.com.mx/help_desk/articulo04.html
43. TURBITT, K., 2006. *ISO 20000: ¿qué deben hacer las organizaciones?* 2006. S.I.: s.n.
44. VISUAL PARADIGM 2010. Visual Paadigm. [en línea]. [Consulta: 20 January 2014]. Disponible en: <http://www.visual-paradigm.com>.
45. VV.AA 2006. *UNE 50-113-91 2 DOCUMENTACION E INFORMACION*. 2006. Madrid, España: s.n. ISBN 2910011565628.

Glosario de Términos

API (Application Programming Interface): Interfaz para programación de aplicaciones. Es un conjunto de funciones y procedimientos que ofrece determinada biblioteca para ser utilizada por otro *software*.

Cliente: Alguien que compra bienes o servicios. El cliente de un proveedor de servicios de TI es la persona o grupo que define y acuerda los objetivos de nivel de servicio.

Diseño del servicio: Es una etapa en el ciclo de vida de un servicio. El diseño del servicio incluye el diseño de los servicios, las prácticas regulatorias, las políticas y procesos requeridos para llevar a cabo la estrategia del proveedor de servicios y facilitar la introducción de servicios en ambientes que tienen soporte.

Framework: Marco de trabajo o conjunto de bibliotecas orientadas a la reutilización a gran escala de componentes de *software* para el desarrollo rápido de aplicaciones.

Gestión de Catálogo de Servicios: Es el proceso responsable de proporcionar y mantener el catálogo de servicios y de asegurar que esté disponible para aquellos que estén autorizados a acceder a él.

Gestión de Niveles de Servicio: Es el proceso responsable de negociar acuerdos de nivel de servicios alcanzables y de asegurar que estos se cumplan.

Gestión de Servicios de la Tecnología de Información: Es la implementación y gestión de la calidad de los servicios de TI que cumplan las necesidades del negocio

ISO/IEC 20000: Una norma internacional de gestión de servicios TI.

ITIL: Es un conjunto de publicaciones de mejores prácticas para la gestión de servicios de

JSP: Surgen con la idea de facilitar la creación de contenido dinámico a desarrolladores sin necesidad de conocer a fondo el lenguaje Java.

Nivel de servicios: Es el logro medido y notificado en informes con respecto a uno o más objetivos de nivel de servicio. A veces se utiliza el término de manera informal en el sentido de objetivo de nivel de servicio.

Tecnología de la Información: Es el uso de la tecnología para el almacenamiento, la comunicación o el procesamiento de la información.

XML: Es un lenguaje de marcado para documentos que contienen información estructurada. Es un mecanismo para identificar las estructuras en un documento.

Anexo 1: Carta de aceptación del cliente



Figura 15: Carta de aceptación del cliente.

Anexo 2: Guía de observación

Objetivo: Comprobar cómo se manifiesta en el Centro de Soporte UCI el desarrollo de los procesos de Gestión de Catálogos y Gestión de Nivel de Servicio.

Nombre: _____

Especialidad: _____

Indicador a tratar: _____

Municipio: _____ **Provincia:** _____

No	Indicadores.
1	Nivel de acceso a la información que se maneja en el departamento.
2	Nivel de importancia de la información que se maneja.
3	Acceso de terceros a la información.

✓ **Indicador 1.**

Alto: Cuando solo los máximos directivos del departamento tienen acceso a la información

Medio: Cuando se asigna permisos de accesos a otra persona en ausencias de los máximos directivos.

Bajo: Cuando acceden todos los trabajadores del centro.

✓ **Indicador 2.**

Alta: Si la información es totalmente confidencial.

Media: Si los especialistas del departamento pueden estar al tanto de la información que se maneja.

Baja: Si la información se encuentra a disposición de todo el personal del centro.

✓ **Indicador 3.**

Alto: Si cualquier personal del centro accede y maneja la información.

Media: Si solo accede a ella el personal autorizado por la dirección del centro.

Baja: Si solo puede manejar la información el personal especializado.

Anexo 3: Entrevista

(realizada al jefe de departamento, especialistas)

Objetivo: Comprobar cómo se realizan los procesos de Gestión de Catálogos y Gestión de Nivel de Servicio

Compañero(a).

Se necesita su valiosa colaboración para la realización de esta investigación, de sus respuestas dependen los resultados de la misma. **MUCHAS GRACIAS.**

Datos generales.

Nombre y Apellidos _____

Años de experiencia _____

Aspectos a encuestar

1. La información que se maneja en los departamentos ¿Es de carácter sensible? ¿Por qué?
2. ¿Cómo se realizan actualmente los procesos de Gestión de Catálogos y Gestión de Nivel de Servicio en el centro?
3. ¿Cualquier persona puede realizar dichos procesos? Argumente.
4. ¿Qué cantidad de personas del departamento tienen acceso a la información que se maneja?
5. ¿Cuál o cuáles son las vías que tiene el cliente para establecer las pautas de los servicios seleccionados?
6. ¿Qué beneficios encuentra usted a la realización de los procesos de Gestión de Catálogos y Gestión de Nivel de Servicio, utilizando un marco de referencia para la ITSM?

Anexo 4: Lista de Reserva del Producto.

Prioridad	Ítem *	Descripción	Estimación	Estimado por	
Muy Alta					
	Insertar Usuario.	1	Permite al administrador insertar un usuario al sistema.	3	Maidelis Vega Calzado, Emil Pérez Piz.
	Modificar Usuario.	2	Permite modificar los datos de un usuario específico.	3	Maidelis Vega Calzado, Emil Pérez Piz.
	Eliminar Usuario.	3	Permite eliminar un usuario específico.	3	Maidelis Vega Calzado, Emil Pérez Piz.
	Mostrar Usuario.	4	Permite mostrar los datos de un usuario específico.	2	Maidelis Vega Calzado, Emil Pérez Piz.
	Insertar Grupo de Usuarios.	5	Permite al administrador insertar un grupo de usuario al sistema.	3	Maidelis Vega Calzado, Emil Pérez Piz.
	Modificar Grupo de Usuarios.	6	Permite al administrador modificar un grupo de usuario existente.	3	Maidelis Vega Calzado, Emil Pérez Piz.
	Eliminar Grupo de Usuarios.	7	Permite al administrador eliminar un grupo de usuario existente.	3	Maidelis Vega Calzado, Emil Pérez Piz.
	Mostrar Grupo de Usuarios.	8	Muestra los datos de un grupo de usuario existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
	Insertar Roles de Usuarios.	9	Permite al administrador insertar un rol de usuario al sistema.	3	Maidelis Vega Calzado, Emil Pérez Piz.
	Modificar Roles de Usuarios.	10	Permite al administrador modificar un rol de usuario existente.	3	Maidelis Vega Calzado, Emil Pérez Piz.
	Eliminar Roles de Usuarios.	11	Permite al administrador eliminar un rol de usuario existente.	3	Maidelis Vega Calzado, Emil Pérez Piz.
	Mostrar Roles de Usuarios.	12	Muestra los datos de un rol de usuario existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
	Insertar Acuerdos de Niveles de Servicios.	13	Permite al administrador insertar un Acuerdo de Nivel de Servicio al sistema.	3	Maidelis Vega Calzado, Emil Pérez Piz.
	Modificar Acuerdos de Niveles de Servicios	14	Permite al administrador modificar los datos de un Acuerdo de .	3	Maidelis Vega Calzado, Emil Pérez Piz.
	Eliminar Acuerdos de Niveles de Servicios.	15	Permite al administrador eliminar un Acuerdo de Nivel de Servicio existente.	3	Maidelis Vega Calzado, Emil Pérez Piz.
	Mostrar Acuerdos de Niveles de Servicios.	16	Muestra los datos de un Acuerdo de Nivel de Servicio existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
	Insertar Servicio.	17	Permite al administrador insertar un Servicio al sistema.	3	Maidelis Vega Calzado, Emil Pérez Piz.
	Modificar Servicio.	18	Permite al administrador modificar los datos de un Servicio existente.	3	Maidelis Vega Calzado, Emil Pérez Piz.

				Pérez Piz.
Eliminar Servicio.	19	Permite al administrador eliminar un Servicio existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Mostrar Servicio.	20	Muestra los datos de un Servicio existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Insertar Áreas Funcionales.	21	Permite al administrador insertar un Área funcional.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Modificar Áreas Funcionales.	22	Permite al administrador modificar los datos de un Área funcional existente.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Eliminar Áreas Funcionales.	23	Permite al administrador eliminar un Área funcional existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Mostrar Áreas Funcionales.	24	Muestra los datos de un Área funcional existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Insertar Acuerdos de Niveles de Operación.	25	Permite al administrador insertar un Acuerdo de Nivel de Operación al sistema.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Modificar Acuerdos de Niveles de Operación.	26	Permite al administrador modificar los datos de un Acuerdo de Nivel de Operación existente.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Eliminar Acuerdos de Niveles de Operación.	27	Permite al administrador eliminar un Acuerdo de Nivel de Operación existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Mostrar Acuerdos de Niveles de Operación.	28	Muestra los datos de un Acuerdo de Nivel de Operación existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Insertar Proveedores.	29	Permite al administrador insertar un nuevo Proveedor al sistema.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Modificar Proveedores.	30	Permite al administrador modificar los datos de un Proveedor existente.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Eliminar Proveedores.	31	Permite al administrador eliminar un Proveedor existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Mostrar Proveedores.	32	Muestra los datos de un Proveedor existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Mostrar Catálogo de Servicios.	33	Muestra los datos de un Catálogo de Servicio existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Alta				
Insertar Evaluación.	34	Permite al administrador insertar una nueva Evaluación al sistema.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Modificar Evaluación.	35	Permite al administrador modificar una Evaluación existente.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Eliminar Evaluación.	36	Permite al administrador eliminar una Evaluación existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Mostrar Evaluación.	37	Muestra los datos de una Evaluación existente.	2	Maidelis Vega Calzado, Emil

				Pérez Piz.
Insertar Urgencia.	38	Permite al administrador insertar una nueva Urgencia al sistema.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Modificar Urgencia.	39	Permite al administrador modificar los datos de una Urgencia existente.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Eliminar Urgencia.	40	Permite al administrador eliminar una Urgencia existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Mostrar Urgencia.	41	Muestra los datos de una Urgencia existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Insertar Impacto.	42	Permite al administrador insertar un nuevo Impacto al sistema.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Modificar Impacto.	43	Permite al administrador modificar los datos de un Impacto existente.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Eliminar Impacto.	44	Permite al administrador eliminar un Impacto existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Mostrar Impacto.	45	Muestra los datos de un Impacto existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Insertar Prioridad.	46	Permite al administrador insertar una nueva Prioridad al sistema.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Modificar Prioridad.	47	Permite al administrador modificar los datos de una Prioridad existente.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Eliminar Prioridad.	48	Permite al administrador eliminar una Prioridad existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Mostrar Prioridad.	49	Muestra los datos de una Prioridad existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Insertar Departamento.	50	Permite al administrador insertar un nuevo Departamento al sistema.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Modificar Departamento.	51	Permite al administrador modificar los datos de un Departamento existente.	3	Maidelis Vega Calzado, Emil Pérez Piz.
Eliminar Departamento.	52	Permite al administrador eliminar un Departamento existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Mostrar Departamento.	53	Muestra los datos de un Departamento existente.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Asociar un Acuerdo de Nivel de Servicio a un Servicio.	54	Permite asociarle a un Servicio determinado un Acuerdo de Nivel de Servicio creado con anterioridad.	1	Maidelis Vega Calzado, Emil Pérez Piz.
Asociar un Acuerdo de Nivel de Operación a un Servicio.	55	Permite asociarle a un Servicio determinado un Acuerdo de Nivel de Operación creado con	1	Maidelis Vega Calzado, Emil Pérez Piz.

		anterioridad.		
Asociar una o más Áreas funcionales a un Servicio.	56	Permite asociarle a un Servicio determinado un Área funcional creada con anterioridad.	1	Maidelis Vega Calzado, Emil Pérez Piz.
Media				
Autenticar Usuario.	57	Le permite al usuario que posee una cuenta entrar al sistema.	1	Maidelis Vega Calzado, Emil Pérez Piz.
Realizar búsquedas simples.	58	Permite realizar búsquedas de servicios, proveedores, áreas funcionales, etc. en el sistema.	1	Maidelis Vega Calzado, Emil Pérez Piz.
Realizar búsquedas avanzadas.	59	Permite realizar búsquedas avanzadas de servicios, proveedores, áreas funcionales, etc. en el sistema por fecha, responsable, entre otros aspectos.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Realizar notificaciones por correos a usuarios.	60	Permite realizar notificaciones por correo a un usuario determinado.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Realizar notificaciones por correos a un grupo de usuarios.	61	Permite realizar notificaciones por correo a un grupo de usuario determinado.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Asociar una evaluación a proveedores.	62	Permite asignarle una evaluación determinada a un proveedor existente.	1	Maidelis Vega Calzado, Emil Pérez Piz.
Realizar reportes de los Acuerdos de Niveles de Servicios y Acuerdos de Niveles de Operación.	63	Permite mostrar los datos de los Acuerdos de Niveles de Servicios y de Operación creados en diferentes formatos.	2	Maidelis Vega Calzado, Emil Pérez Piz.
Adjuntar un documento a los Acuerdos de Niveles de Servicios.	64	Permite adjuntar un documento previamente creado a un Acuerdo de Nivel de Servicio determinado.	1	Maidelis Vega Calzado, Emil Pérez Piz.
Adjuntar un documento a los Servicios.	65	Permite adjuntar un documento previamente creado a un Servicio determinado.	1	Maidelis Vega Calzado, Emil Pérez Piz.
Mostrar últimos Acuerdos de Niveles de Servicios creados.	66	Permite mostrar los datos de los últimos Acuerdos de Niveles de Servicios creados en el sistema.	1	Maidelis Vega Calzado, Emil Pérez Piz.
Mostrar últimos Acuerdos de Niveles de Operación creados.	67	Permite mostrar los datos de los últimos Acuerdos de Niveles de Operación creados en el sistema.	1	Maidelis Vega Calzado, Emil Pérez Piz.
Mostrar matriz de asignación de prioridad.	68	Permite mostrar una matriz con todas las prioridades creados en el sistema y su nivel correspondiente.	1	Maidelis Vega Calzado, Emil Pérez Piz.
Mostrar información del sitio (manual de ayuda).	69	Permite mostrar el manual de ayuda del sitio, para que el usuario pueda navegar con mayor facilidad.	1	Maidelis Vega Calzado, Emil Pérez Piz.
Mostrar reportes gráficos.	70	Permite mostrar mediante un gráfico de pastel la cantidad de Acuerdos de Niveles de Servicios, de Operación, Servicios y Áreas funcionales creadas.	1	Maidelis Vega Calzado, Emil Pérez Piz.
Requisitos No Funcionales				
Ítem	Descripción			
1	Usabilidad: Describen los niveles apropiados de usabilidad para los usuarios finales del producto.			
2	Diseño e Implementación: Especifica o restringe la codificación o construcción de un sistema, son restricciones que deben ser cumplidas estrictamente.			

3	Funcionamiento: Se describen los requisitos de <i>software</i> y hardware que debe presentar el ordenador.
4	Seguridad: Sólo el personal autorizado deberá hacer uso del sistema, entiéndase personal autorizado los especialistas del Centro de Soporte UCI.
5	Fiabilidad: Se debe tener en cuenta la recuperación frente a fallos de conexión, asegurando que no se pierdan los datos del perfil definido por el usuario.
6	Interfaz de Usuario: Se describe la apariencia del producto. Se especifica cómo se pretende que sea la interfaz externa del producto

Tabla 21: Lista de reserva del producto.

Anexo 6: Prueba de rendimiento

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Etiqueta	# Muestras	Media	Mediana	Línea de 9...	Mín	Máx	% Error	Rendimiento	Kb/sec
/aplica...	6	4	4	4	3	7	0,00%	50,7/min	1,0
/aplica...	6	4	4	5	3	7	0,00%	51,3/min	1,0
/aplica...	6	4	4	5	3	5	0,00%	51,3/min	1,0
/aplica...	6	4	4	5	3	6	0,00%	51,7/min	1,0
/aplica...	6	4	4	6	2	7	0,00%	51,9/min	1,0
/aplica...	6	81	4	8	3	463	0,00%	1,1/sec	1,2
/aplica...	6	82	4	238	3	243	0,00%	1,4/sec	1,6
/aplica...	3	289	263	537	67	537	0,00%	35,0/min	11,6
/aplica...	5	49	8	10	7	217	0,00%	50,2/min	1,0
/aplica...	5	9	10	10	4	12	0,00%	50,9/min	1,0
/aplica...	3	7	7	9	7	9	0,00%	41,5/min	,8
/aplica...	5	52	7	7	4	237	0,00%	58,6/min	1,2
/aplica...	5	111	7	246	5	291	0,00%	1,2/sec	1,4
/aplica...	5	8	7	10	6	12	0,00%	1,4/sec	1,6
/aplica...	4	4	4	5	4	5	0,00%	2,0/sec	2,3
/aplica...	4	5	5	6	5	6	0,00%	1,9/sec	2,3
/aplica...	4	4	4	5	4	5	0,00%	1,8/sec	2,2
/aplica...	4	58	4	223	4	223	0,00%	2,0/sec	2,4
/aplica...	4	5	4	11	3	11	0,00%	2,1/sec	2,5
/aplica...	4	4	4	4	4	4	0,00%	2,2/sec	2,6
/aplica...	4	4	4	5	3	5	0,00%	2,2/sec	2,6
/aplica...	4	3	3	4	3	4	0,00%	2,1/sec	2,5
/aplica...	4	4	4	5	3	5	0,00%	2,1/sec	2,5
/aplica...	4	4	4	5	3	5	0,00%	2,1/sec	2,5
/aplica...	4	4	4	5	4	5	0,00%	2,1/sec	2,5
/aplica...	4	3	3	4	3	4	0,00%	2,3/sec	2,8
/aplica...	4	4	4	5	3	5	0,00%	2,3/sec	2,7
Total	10536	546	16	606	2	145837	0,00%	5 9/sec	927,4

¿Incluir el nombre del grupo en la etiqueta? Guardar la cabecera de la tabla

Figura 17: Prueba de rendimiento para 300 peticiones concurrentes.

Anexo 7: Vistas de la solución propuesta.

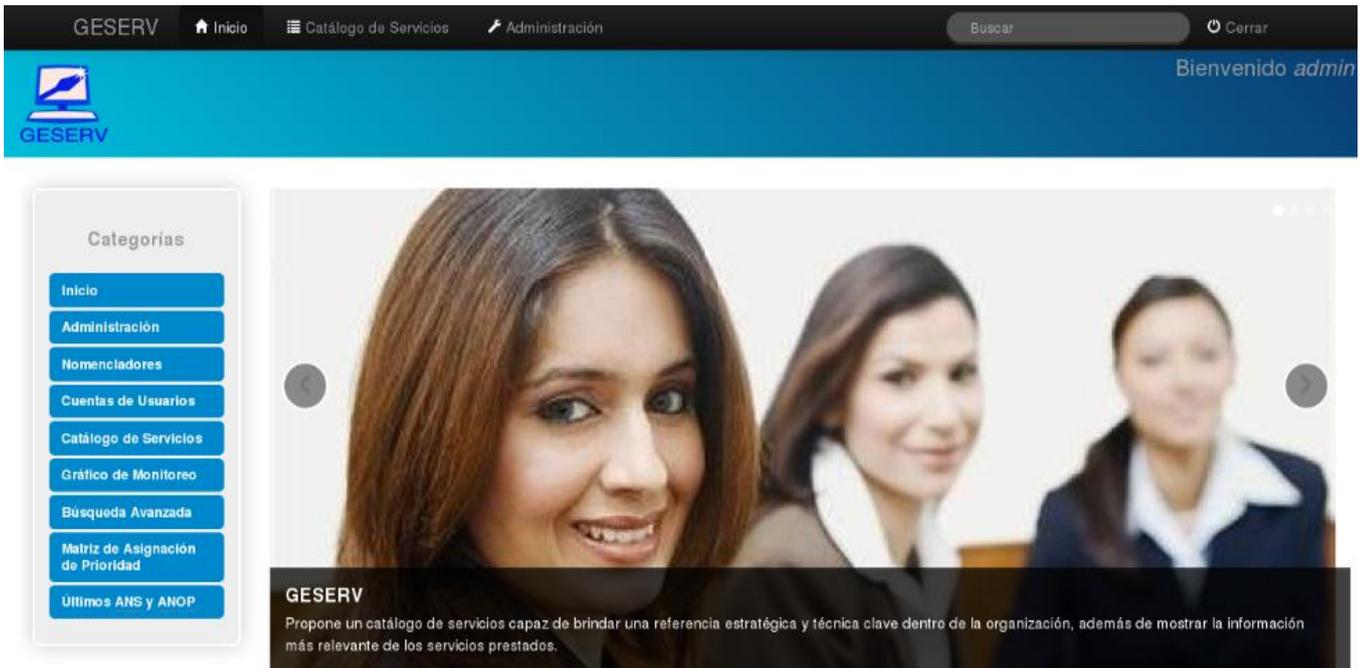


Figura 18: Vista principal de la solución propuesta.



Figura 19: Vista administrativa de la solución propuesta.



Figura 20: Gestión de cuentas de usuario de la solución propuesta.