



Universidad de las Ciencias Informáticas

Facultad 2

“Módulo de Gestión de Eventos para el Centro de
Soporte de la Universidad de las Ciencias
Informáticas”

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

Autores: Amanda Gómez Marrero
Tatiana Miranda Terrero

Tutores: Msc. Yulio Seriocha García Gallardo
Ing. Neybis Lago Clara

La Habana, 27 de Junio de 2014
“Año 56 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los 27 días del mes de junio del año 2014.

Autores:

Amanda Gómez Marrero

Tatiana Miranda Terrero

Tutores:

Msc. Yulio Seriocha García Gallardo

Ing. Neybis Lago Clara

Datos de Contacto

Tutores:

- Msc. Yulio Seriocha García Gallardo, graduado de Ingeniero en Ciencias Informáticas de la UCI del año 2007, se desempeña como especialista del Centro Soporte UCI. Es máster en informática aplicada. Categoría docente: Asistente.(ygarcia@uci.cu)
- Ing. Neybis Lago Clara, graduada de Ingeniero en Ciencias Informáticas de la UCI del año 2012, se desempeña como especialista del Centro Soporte UCI.(nlago@uci.cu)

Dedicatoria

Este trabajo está dedicado especialmente a mi mamá y a mi papá, por ser las personas más maravillosa de mi mundo, ellos saben que este título es de ellos, los amo.

Tatiana.

A Dios por ser el primero en permitir que yo exista y darme las fuerzas para culminar mis estudios universitarios.

Mis padres quienes lo han dado todo por favorecer mis tiempos de estudios y han estado conmigo en todo tiempo.

Mi hermana Adriadna, quien ha dado mucho de sí para mi, por su paciencia y ser mi tutora #1 en estos años.

Y por último y no menos importante, mi sobrino Daniel, quien le debo estos últimos 6 meses de despeje, alegría y ser una bendición super especial para mi familia.

Con amor sincero a ustedes quienes son los 1ros en mi vida

Amanda.

Agradecimientos

A Diosito por darme la dicha y la salud para llegar a ser lo que soy hoy y por regalarme la familia más linda del mundo.

A mi Virgencita de la Caridad del Cobre por darme la protección que siempre me acompaña.

A mi mamá que sé que siempre está conmigo, cuidándome y guiándome en la vida, te amo mucho.

A mi mamá, gracias por la fuerza, la dedicación y la inspiración para seguir adelante siempre.

A mi papá, gracias por la confianza y la perseverancia puesta en mi y por ser un ejemplo a seguir en todos los aspectos de la vida.

A mi tía Tania por ser mi tercera mamá, por ayudarme siempre y estar al pendiente mío, te quiero mucho.

A mi abuelo Ino por estar al pendiente siempre de mi educación.

A mi hermano, por ser el único que tengo y aunque no se lo digo mucho él es muy importante en mi vida, te quiero.

A mi tío Julio por ser el tío más especial que tengo.

A Sergio por ser el complemento de mi vida y por dedicarse a mi todo este tiempo.

Y a todos mis amigos por ser tan especiales conmigo y soportarme en su vida, los quiero muchísimo.

A nuestros tutores por su entrega y dedicación durante el desarrollo de este trabajo, gracias.

Tatiana.



Al Creador de los cielos y la tierra, al que le da sustento a todo lo que existe, el que ha permitido mi existencia y es mi guardador, ayudador, mi torre fuerte, el que sostiene mi mano derecha y me dice: "no temas, yo te ayudo". El que me ha permitido disfrutar cada día de mi vida con nuevas fuerzas, con fe, esperanza, gozo y con un inagotable amor que reboza desde mi familia hasta cada ser querido que ocupa un lugar especial en mi corazón: Mi Dios.

A mi familia, porque allí siento el calor de Dios y conozco lo que es amar sin condición alguna. Allí con mis padres que me han formado una mejor persona para saber tomar decisiones en mi vida, para valerme por mí misma, y si tengo alguna caída no faltará su pronta ayuda para levantarme, animarme, sanarme y fortalecerme nuevamente para continuar en este curso de la vida.

Mi mami, por cada lágrima derramada en mi ausencia, por su amor único que le caracteriza a una madre, pero la mía especial, por la fuerza interna que tiene para seguir conmigo (que no hay quien la iguale) y por tus incansables consejos e instrucciones. No me parece merecerla, pero Dios me ha placido tenerla.

Mi papi, quien ha sabido ser ejemplo para mi vida, amigo, mejor confidente, de quien siempre he recibido su mano para ser mi ayudador. Es quien sabe cómo hacerme reír, sentirme confiada, segura, que no tengo que temer porque él estará siempre conmigo.

Mi tata linda (Adriadna) quien es la mejor hermana del mundo, por ser mi mejor amiga, por la relación tan especial que desde bien pequeñas hemos sabido cuidar. Por ser en momentos mi madre, en otros mi payasita y en otros mi maestra.

A mi pastor Carlos Sebastián y su esposa Olga, por sus incansables atenciones para conmigo, sus oraciones y ser líderes en mi vida.

Alberto y Mairim, por ser amigos de calidad, por estar conmigo en buenas y en malas, siempre mostrando su ayuda y nunca faltando el calor de sus abrazos.

Al Dpto. de Jóvenes de la Iglesia del Cotorro, por ser mi otra familia, por su cariño, atención, respeto y me hacen sentir especial, porque estoy entre personas especiales, atentas y de calidad.

Mis amigos Daniel y Yazmin quienes han estado conmigo estos 5 años, siendo como hermanos, recibiendo de ellos amor, respeto, comprensión, aceptación, lealtad, porque también han estado conmigo en buenas y malas y siempre las llevaré conmigo.

Mario, por su paciencia, atención, por la amistad transparente que me ha brindado, por saber escucharme cuando lo he necesitado, por cada tiempo que me ha dedicado y por los tiempos de estudios que me facilitó en su laptop.

A mis compañeros de grupo, por todo este tiempo de estudios, en el que nos hemos dado ánimo unos a otros, por todo este tiempo de muchos estudios juntos. En especial Tatiana (compañera de tesis), Damisel, Yarianna (compañera de cuarto) y Lisset quienes hemos estado juntas desde 1er año.

A mis tutores, por su dedicación a nosotras y tener un excelente comportamiento, atención y defensa a favor de culminar de manera satisfactoria nuestro trabajo de Diploma.

A todos Gracias.

Amanda

RESUMEN

La Biblioteca de Infraestructura de las Tecnologías de la Información (ITIL) es un estándar ampliamente conocido para la Gestión de Servicios de las Tecnologías de la Información (TI). Uno de los procesos asociados a la Gestión de Servicio es la Gestión de Eventos, que es el responsable de monitorizar todos los eventos que acontezcan en la infraestructura TI.

La plataforma que se usa actualmente en el Centro de Soporte UCI¹ no gestiona el proceso de Gestión de Eventos definido por ITIL. Por este motivo surge la necesidad de desarrollar un módulo que controle el proceso de Gestión de Eventos de la metodología ITIL 2011 de los servicios que presta el Centro de Soporte UCI. El mismo fue desarrollado sobre la plataforma Java, con el lenguaje de programación Groovy, utilizando el *framework* Grails 2.1. Se empleó la metodología de desarrollo XP² y MVC³ como patrón de arquitectura.

La validación del módulo desarrollado se realizó mediante pruebas unitarias y de aceptación, identificándose un conjunto de no conformidades que fueron corregidas en varias iteraciones. La realización de estas pruebas arrojó como resultado que el *software* está listo para integrarse al Sistema de Servicios del Centro de Soporte UCI.

Palabras claves:

Gestión de Evento, metodología ITIL 2011, servicios de TI, tecnologías de información.

¹ Universidad de las Ciencias Informáticas.

² Programación Extrema(*eXtreme Programming*, por sus siglas en inglés)

³ Modelo – Vista – Controlador.

Índice

INTRODUCCIÓN 13

CAPÍTULO 1. FUNDAMENTOS TEÓRICOS METODOLÓGICOS RELACIONADOS CON LA GESTIÓN DE EVENTOS..... 17

1.1 PRINCIPALES CONCEPTOS ASOCIADOS A LA GESTIÓN DE EVENTOS 17

1.2 ITIL 18

 1.2.1 Integración de estándares 20

1.3 PROCESO DE GESTIÓN DE EVENTOS 22

1.4 SISTEMAS HOMÓLOGOS..... 24

1.5 HERRAMIENTAS Y TECNOLOGÍAS. 25

 1.5.1 Metodología de Desarrollo XP 25

 1.5.2 Patrón Arquitectónico: MVC 26

 1.5.3 Herramienta CASE: Visual Paradigm 8.0 26

 1.5.4 Framework: Grails 2.1 26

 1.5.5 Lenguaje de Programación: Groovy 1.8..... 27

 1.5.6 IDE de Desarrollo: IntelliJ IDEA 11.0..... 27

 1.5.7 Sistema de Gestión de Bases de Datos: PostgreSQL9.1..... 28

1.6 CONCLUSIONES..... 28

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN DEL MÓDULO DE GESTIÓN DE EVENTOS..... 29

2.1 SOLUCIÓN PROPUESTA..... 30

2.2 ROLES Y RESPONSABILIDADES 31

2.3 REQUISITOS DEL SISTEMA 32

 2.3.1 Requisitos Funcionales..... 32

 2.3.2 Requisitos No Funcionales 32

2.4 HISTORIAS DE USUARIO 36

2.5 PLAN DE ITERACIONES 39

2.6 TARJETA CRC..... 43

2.7 PATRONES DE DISEÑO..... 45

 2.7.1 Patrón de diseño..... 45

2.7.2 Patrón de Arquitectura	48
2.8 MODELO CONCEPTUAL	29
2.9 DISEÑO DE BASE DE DATOS	53
2.10 ESTÁNDAR DE CODIFICACIÓN	54
2.11 CONCLUSIONES.....	55
CAPÍTULO 3. VALIDACIÓN DEL MÓDULO DE GESTIÓN DE EVENTOS.....	57
3.1 PRUEBAS UNITARIAS	57
3.2 PRUEBAS DE ACEPTACIÓN	61
3.2 CONCLUSIONES.....	64
CONCLUSIONES.....	65
RECOMENDACIONES.....	66
REFERENCIAS BIBLIOGRÁFICAS.....	67
BIBLIOGRAFÍA.....	70
ANEXOS	73
ANEXO 1. HISTORIAS DE USUARIOS.....	73
ANEXO 2. TARJETAS CRC.	79
ANEXO 3. PRUEBAS DE ACEPTACIÓN.....	84
ANEXO4. ENTREVISTA.	89
ANEXO5. AVAL DEL CLIENTE.....	91
GLOSARIO DE TÉRMINOS	93
 Índice Tablas	
TABLA 1.ROLES Y RESPONSABILIDADES	31
TABLA 2. REQUISITOS FUNCIONALES DEL MÓDULO	32
TABLA 3. HU AUTENTICAR USUARIO.	38
TABLA 4. HU GESTIONAR EVENTO.....	39
TABLA 5. HU GESTIONAR USUARIO.....	42
TABLA 6. PLAN DE ITERACIONES	43

TABLA 7. TARJETA CRC#2 EVENTOCONTROLLER	44
TABLA 8. TARJETA CRC#6 USUARIOCONTROLLER.	45
TABLA 9. PRUEBA DE ACEPTACIÓN#1 AUTENTICAR USUARIO.	63
TABLA 10. PRUEBA DE ACEPTACIÓN#2 GESTIONAR EVENTO.....	63
TABLA 11. PRUEBA DE ACEPTACIÓN#3 GESTIONAR USUARIO.....	64
TABLA 12. HU#3 GESTIONAR GRUPO DE USUARIOS.....	75
TABLA 13. HU#4 GESTIONAR CATEGORÍA.....	77
TABLA 14. HU#5 GESTIONAR SUB_CATEGORÍA	79
TABLA 15. TARJETACRC#3	79
TABLA 16. TARJETA CRC#4.....	80
TABLA 17. TARJETA CRC#5.....	81
TABLA 18. TARJETA CRC#7.....	81
TABLA 19. TARJETA CRC#8.....	82
TABLA 20. TARJETACRC#9	82
TABLA 21. TARJETA CRC#10.....	82
TABLA 22. TARJETA CRC#11.....	83
TABLA 23. TARJETA CRC#12.....	83
TABLA 24. TARJETA CRC#13.....	84
TABLA 25. PRUEBA DE ACEPTACIÓN #3	84
TABLA 26. PRUEBA DE ACEPTACIÓN #4.....	85
TABLA 27. PRUEBA DE ACEPTACIÓN #5.....	85
TABLA 28. PRUEBA DE ACEPTACIÓN #7.....	86
TABLA 29. PRUEBA DE ACEPTACIÓN #8.....	86
TABLA 30. PRUEBA DE ACEPTACIÓN#9	87
TABLA 31. PRUEBA DE ACEPTACIÓN #11.....	88
TABLA 32. PRUEBA DE ACEPTACIÓN #12.....	88
TABLA 33. PRUEBA DE ACEPTACIÓN #13.....	89
TABLA 34. PRUEBA DE ACEPTACIÓN #15.....	89

Índice de Ilustraciones

ILUSTRACIÓN 1. FASES DE ITIL 2011 19

ILUSTRACIÓN 2. FLUJO DE GESTIÓN DE EVENTO..... 23

ILUSTRACIÓN 3. MODELO CONCEPTUAL 30

ILUSTRACIÓN 4. EJEMPLO DEL PATRÓN EXPERTO..... 46

ILUSTRACIÓN 5. EJEMPLO DEL PATRÓN CREADOR 47

ILUSTRACIÓN 6.EJEMPLO DEL PATRÓN CONTROLADOR..... 48

ILUSTRACIÓN 7. ESTRUCTURA DEL PATRÓN DE ARQUITECTURA MVC 49

ILUSTRACIÓN 8. EJEMPLO DEL COMPONENTE MODELO..... 50

ILUSTRACIÓN 9. EJEMPLO DEL COMPONENTE VISTA 51

ILUSTRACIÓN 10. EJEMPLO DEL COMPONENTE VISTA DESDE LA WEB..... 52

ILUSTRACIÓN 11. EJEMPLO DEL COMPONENTE CONTROLADOR..... 53

ILUSTRACIÓN 12. DISEÑO DE LA BASE DE DATOS..... 54

ILUSTRACIÓN 13. EJEMPLO DEL USO DEL CAMEL CASE 55

ILUSTRACIÓN 14. DIAGRAMA DE DESPLIEGUE DEL MÓDULO GESTIÓN DE EVENTOS 56

ILUSTRACIÓN 15. GRAFO DEL FLUJO ASOCIADO AL MÉTODO EVENTOS ()..... 58

ILUSTRACIÓN 16. GRAFO DEL FLUJO ASOCIADO AL MÉTODO SAVE ()..... 59

ILUSTRACIÓN 17. GRAFO DEL FLUJO ASOCIADO AL MÉTODO JSONDATA () 60

INTRODUCCIÓN

Las Tecnologías de la Información, en lo adelante TI, han jugado un papel importante en el transcurso de la historia, y han llegado a convertirse en una herramienta imprescindible y clave para empresas e instituciones. Estas se entienden como: *"aquellas herramientas y métodos empleados para recabar, retener, manipular o distribuir información"* (1).

Las empresas utilizan las TI para gestionar casi todos los aspectos del negocio, especialmente el manejo de los registros financieros y transaccionales de las organizaciones (2). Cuando estos servicios se vuelven críticos, cada una de las actividades que se realizan debe ejecutarse con un orden determinado, es ahí donde la Gestión de Servicios de las TI representa un elemento crucial para toda empresa que persiga obtener una ventaja competitiva y mejorar la calidad de sus productos.

La Gestión de Servicios de las TI es una disciplina basada en procesos, enfocada en alinearlos de forma proporcional, con las necesidades de las empresas. Propone cambiar el modelo de gestión de TI, por una colección de componentes enfocados en servicios, usando distintos marcos de trabajo que incluyen un conjunto de buenas prácticas.

Los objetivos de una buena Gestión de Servicios han de ser, proporcionar una adecuada gestión de la calidad y aumentar la eficiencia (3). Además alinear los procesos del negocio y la infraestructura TI, reducir los riesgos asociados a los Servicios TI y generar el negocio.

La tendencia de Gestión de Servicio TI se basa en la promoción y soporte de aplicación de las mejores prácticas, marcos referenciales y estándares de aceptación internacional, tales como Organización Internacional de Normalización / Comisión Electrotécnica Internacional (ISO/IEC 20000), ITIL, Objetivos de Control para la Tecnología de la Información (COBIT) y el Modelo de Madurez de la Capacidad Integrado para Servicios (CMMI-SVC).

Un estándar conocido para la Gestión de los Servicios de las TI es ITIL. Este se centra en brindar servicios de alta calidad para lograr la máxima satisfacción del cliente a un costo manejable, en esencia, determina la forma de ejecutar procesos ayudados de la tecnología para lograr la satisfacción de los usuarios (11). El objetivo de ITIL es proporcionar a los administradores de sistemas de las TI las mejores herramientas y documentos que les permitan mejorar la calidad de sus servicios, es decir, mejorar la satisfacción del cliente al mismo tiempo que alcanzan los objetivos estratégicos de la empresa.

En la Gestión de Servicios, lo que está relacionado estrechamente con ciertas partes de la infraestructura de las TI, son exactamente los Servicios de Soporte. Estos se preocupan de todos los aspectos que garanticen la continuidad, disponibilidad y calidad del servicio prestado al usuario (4). Son estos servicios, mediante el cual los especialistas en apoyo informático, proporcionan asistencia técnica, soporte remoto y asesoramiento a individuos y organizaciones que dependen de las TI.

Uno de los procesos asociados directamente a la Gestión de Servicios es la Gestión de Eventos. Este es el responsable de “monitorizar todos los eventos que acontezcan en la infraestructura TI con el objetivo de asegurar su correcto funcionamiento y ayudar a prever incidencias futuras” (5). Algunas de las ventajas que una correcta Gestión de Eventos aporta a la organización TI es la ayuda a la detección temprana de incidencias, llegando a evitar que éstos se manifiesten a los usuarios. Además la coordinación directa con otros procesos hace posible que éstos reaccionen con mayor rapidez.

Alrededor del mundo existen varias empresas que se dedican al desarrollo de soluciones informáticas y productos de software. Cuba no está ajena a estos avances tecnológicos y desde hace varias décadas viene desarrollando una estrategia encaminada al desarrollo de la industria del software, por ello se han creado empresas e instituciones como la Universidad de las Ciencias Informáticas (UCI).

La UCI es una de las principales instituciones productoras de software del país, cuenta con varios centros de desarrollo orientados a la aplicación de la informática en las diferentes esferas de la sociedad. Cuenta además con un Centro de Soporte que ofrece servicios de soporte y mantenimiento a los productos desarrollados en los distintos centros de investigación de la institución, y resuelve a los clientes las incidencias tecnológicas que tengan solución inmediata al alcance del equipo técnico.

Actualmente en el Centro de Soporte no se registran los eventos que se originan cuando en un servicio o dispositivo se ha producido un fallo o simplemente han existido cambios en el sistema y puede llegar a convertirse en una incidencia. Tampoco cuenta con una base de datos que almacene dichos eventos y que permita que los especialistas, proveedores, clientes o cualquier involucrado pueda revisar el seguimiento correcto del servicio brindado.

El desconocimiento de la ocurrencia de una actualización en los productos informáticos a través de la Gestión de Cambio y la realización de modificaciones en la documentación que generan los contratos de servicios, son factores que manifiestan la necesidad del correcto funcionamiento de este proceso dentro de la organización de las TI. Debido a esto, se incumple en los tiempos de respuesta y resolución establecidos

con el cliente y los especialistas se demoran en determinar la causa raíz y los síntomas ante un problema detectado. Además se desconocen los cambios realizados a las aplicaciones informáticas, lo que provoca demoras en el servicio a los clientes.

Por lo antes planteado, se identifica como **problema a resolver**: ¿Cómo gestionar los eventos de los servicios en el Centro de Soporte UCI? Definiéndose como **objeto de estudio**: el proceso de Gestión de Eventos según la metodología ITIL 2011. Enmarcándose como **campo de acción**: la Gestión de Eventos según la metodología ITIL 2011 en el Centro de Soporte UCI.

De acuerdo con la problemática planteada se propone como **objetivo general** desarrollar un módulo que gestione los eventos de los servicios que presta el Centro de Soporte UCI según la metodología ITIL 2011.

Para dar cumplimiento al objetivo planteado se proponen las siguientes **tareas de la Investigación**:

1. Realización de un análisis crítico y valorativo de los sistemas informáticos existentes a nivel internacional para la Gestión de Eventos.
2. Realización del estudio de las herramientas, tecnologías y metodología de desarrollo a utilizar.
3. Elaboración de los artefactos establecidos por la metodología de desarrollo a utilizar.
4. Implementación del módulo propuesto, siguiendo lo establecido por los requisitos del *software*.
5. Validación de la solución propuesta mediante el diseño y ejecución de pruebas unitarias y de aceptación.

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

Teóricos:

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

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

Empíricos:

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

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

Capítulo 1: Fundamentos teóricos metodológicos relacionados con la Gestión de Eventos.

Se definen elementos teóricos metodológicos que sustentan la investigación, se hace referencia a los conceptos fundamentales para permitir entender cualquier concepto asociado a la comprensión del trabajo, se realiza un estudio del estado del arte de las soluciones existentes que puedan dar salida de alguna forma al problema científico planteado. El análisis de las diferentes metodologías y tecnologías que se emplean en la solución para llevar a cabo un proceso con calidad en el desarrollo de la investigación.

Capítulo 2: Propuesta de solución del Módulo de Gestión de Eventos.

Se realiza la especificación de los requisitos funcionales y no funcionales del módulo, según necesidades del cliente, y a partir de estos el módulo es diseñado e implementado, para darle solución a la situación problemática planteada.

Capítulo 3: Validación del Módulo de Gestión de Eventos.

Aborda todo lo relacionado con la validación del módulo. Se recoge toda la documentación relacionada con las pruebas realizadas al subsistema, los estándares de codificación y patrones de diseño. Además se muestra y describe el método con el que se realizan las pruebas.

CAPÍTULO 1. FUNDAMENTOS TEÓRICOS METODOLÓGICOS RELACIONADOS CON LA GESTIÓN DE EVENTOS.

En el presente capítulo se exponen los conceptos fundamentales asociados al dominio del problema planteado, con el objetivo de analizar en el marco teórico el tema del objeto de estudio. Se realiza el análisis del estado del arte de los diferentes sistemas homólogos encontrados a nivel internacional, conociendo sus ventajas y desventajas. Se especifican las principales características de la metodología de Gestión de Servicio ITIL 2011, específicamente el proceso de Gestión de Eventos. Se abordan además, las principales características de las tecnologías y herramientas que serán utilizadas para la implementación de la solución propuesta.

1.1 Principales conceptos asociados a la Gestión de Eventos

Con el objetivo de alcanzar una mejor claridad de las bases conceptuales relacionadas con la Gestión de Eventos, a continuación se definen los principales conceptos asociados con el desarrollo de la presente investigación.

Servicio

Según el estándar ITIL, servicio se define como *“medio para entregar valor a los clientes facilitándoles un resultado deseado sin la necesidad de que estos asuman los costos y riesgos específicos asociados”* (6). Lo definen Lamb, Hair y McDaniel como el *“resultado de la aplicación de esfuerzos humanos o mecánicos a personas u objetos. Los servicios se refieren a un hecho, un desempeño o un esfuerzo que no es posible poseer físicamente”* (7).

Para Ivan Thompson, servicio significa las *“actividades identificables que son el resultado de esfuerzos humanos o mecánicos que producen un hecho, que implican generalmente la participación del cliente y que no es posible poseer físicamente, ni transportarlos o almacenarlos”* (8).

Teniendo en cuenta las definiciones anteriores propuestas, para esta investigación se define servicio como el conjunto de actividades que desarrolla una empresa o institución, enfocadas en satisfacer las necesidades de los clientes.

Evento

A efectos de la operación de servicio, se denomina evento a todo suceso detectable que tiene importancia para la estructura de la organización TI, para la prestación de un servicio o para la evaluación del mismo (5). Ejemplos típicos de eventos son las notificaciones creadas por los servicios, los elementos de configuración (CI)⁴ o las herramientas de monitorización y control.

ITIL 2011 lo define como el *“cambio de estado que tiene importancia para la gestión de servicios u otro CI. El término también se utiliza en el sentido de una alerta o notificación creada por cualquier servicio de TI”* (9).

Teniendo en cuenta las definiciones anteriores se define evento a todo acontecimiento previsto o imprevisto que es detectado y notificado por cualquier servicio de TI, con los cuales se toman acciones y se registran los incidentes.

1.2 ITIL

ITIL es un estándar que se centra en brindar servicios de alta calidad para lograr la máxima satisfacción del cliente a un costo manejable. Para ello, parte de un enfoque estratégico basado en el triángulo procesos-personas-tecnología (11). La Gestión de Servicios con ITIL tiene su columna vertebral en la función de Service Desk⁵, la cual es el punto único de contacto entre la organización y el cliente del servicio.

Actualmente existe la nueva edición ITIL 2011 que fue publicada en agosto de 2011. Esta consolida el modelo de ciclo de vida del servicio separando y ampliando algunos subprocesos hasta convertirlos en procesos especializados. Esta modificación responde a un enfoque empresarial para grandes corporaciones que utilizan ampliamente ITIL en sus operaciones. Incluye concretamente cinco fases: Estrategias de los Servicios, Diseño de Servicios, Transición de los Servicios, Operación de los Servicios, y Mejora Continua; enfocándose la investigación en la Gestión de Eventos que pertenece a la Operación de los Servicios, como se muestra en la Ilustración 1.

⁴ Tantos los servicios que ofrecen los servicios de las TI.

⁵ Escritorio de Servicio.



Ilustración 1. Fases de ITIL 2011

ITIL define un flujo de trabajo y un conjunto de prácticas orientadas a procesos, y principalmente está presente en el ciclo de vida completo del servicio. Además posibilita realizar la integración con diferentes estándares como: COBIT, ISO 20000 y CMMI, en caso de que se necesite. Esta metodología es de dominio público ya que cualquiera puede usarlo y es aceptado en todo el mundo como guía para administrar servicios TI. Además se adapta a todos los sectores de la organización sin importar el tamaño de las mismas. Estas y otras características permiten su uso en la implementación del módulo a desarrollar.

Otros de los beneficios (11) del uso de esta metodología son:

1. Mayor alineamiento de las TI con el negocio.
2. Resolución de incidencias y problemas más rápida y eficiente.
3. Aumento de la resolución de incidencias en primera instancia.
4. Implantación de cambios más rápida y mejor control de cambios.
5. Reducción del número de cambios que necesiten ser revocados.

1.2.1 Integración de estándares

Es necesario el uso de normas, estándares y buenas prácticas a la hora de gestionar los servicios de las TI, ejemplo de estas normas son: COBIT, CMMI-SVC, ISO/IEC 20000 e ITIL. Existen muchos estándares pero no todos pueden usarse en conjunto debido a sus diferencias. La razón principal por la cual se realiza una integración es para ayudar a la organización a cumplir sus objetivos de negocios (11).

Es de suma importancia dominar que parte de cada estándar puede ser para cada empresa u organización. Por ello, es fundamental el saber elegir las mejores prácticas, procesos y estrategias entre todos estos modelos y poder generar a partir de ellas un modelo personalizado y adaptado totalmente para una organización en particular. A continuación se muestra como se integran cada uno de los estándares antes mencionado con ITIL.

COBIT

Es un estándar orientado a la Gobernabilidad en las TI. Surge como una alternativa factible para ser utilizada como una guía de acción al momento de garantizar la calidad de los procesos relacionados con el monitoreo, control, calidad y seguridad de los datos correspondientes a transacciones contables. Es una herramienta gerencial que le permite a las organizaciones grandes o pequeñas administrar la información de manera efectiva, eficiente, íntegra y confiable, bajo el estudio de los procesos y actividades, utilizando los recursos de las TI, dígame datos, aplicaciones, tecnología y el recurso humano, con la finalidad de satisfacer los objetivos del negocio, en donde la información necesita cumplir con ciertos criterios, a los cuales COBIT le llama, requerimientos de negocio para la información (10).

Integración de ITIL con COBIT.

COBIT presenta puntos de confluencia con ITIL ya que abarca todo el espectro de actividades de las TI. Ambos modelos son también complementarios y se pueden usar juntos: ITIL para lograr efectividad y eficiencia en los servicios de las TI 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 (11).

ISO/IEC 20 000.

Es una norma que está orientada a procesos pero se enfoca en la calidad. Es un estándar internacional de las TI dirigida específicamente a su Gestión de Servicios. Fue desarrollada en respuesta a la necesidad de

establecer procesos y procedimientos para minimizar los riesgos en los negocios provenientes de un colapso técnico del sistema de las TI de las organizaciones. ISO / IEC 20000 fue lanzado en 2005 sobre la base de ITIL (12).

Integración de ITIL con ISO/IEC 20 000.

ISO/IEC 20000 está alineada con el enfoque basado en procesos definidos en ITIL, siendo así totalmente compatible con esta. En la ISO/IEC 20000, las organizaciones deben ser auditadas y medidas frente a un conjunto establecido de requisitos.

CMMI-SVC.

El modelo CMMI es una asociación de modelos de mejora de procesos e ingeniería del software. Establece una manera de medir el grado de madurez de las organizaciones respecto a la aplicación de las mejores prácticas de desarrollo y gestión del software, así como la prestación de servicios de las TI. CMMI-SVC es una guía para proporcionar servicios en una organización y a clientes externos. Esta norma se basa en conceptos y prácticas de CMMI y otros estándares enfocados en servicios como COBIT, ISO/IEC 20 000 e ITIL. Cuenta con 5 niveles de madurez (11).

Integración de ITIL con CMMI-SVC.

CMMI generalmente se aplica al desarrollo del servicio o infraestructura en las TI (durante la implantación). Este e ITIL tienen un punto común de intersección: La gestión de la entrega. Ambos modelos poseen actividades recomendadas para la gestión de la entrega de nuevos elementos de software e infraestructura.

CMMI se centra en garantizar la calidad en el desarrollo de software mientras que ITIL garantiza la explotación del producto software. Por ello, muchas empresas consideran que ambas metodologías no son excluyentes, sino complementarias, embarcándose en proyectos de análisis y definición de procesos que permitan encajar ambas filosofías de trabajo. En conjunto abarcan desde el desarrollo del software hasta la gestión del mantenimiento y servicios del mismo (11).

Análisis Crítico.

Teniendo en cuenta la integración de ITIL con los diferentes estándares para el Centro de Soporte, es factible solamente usar como estándar a ITIL, sin necesidad de integrarlo, ya que se centra en los procesos

de servicios y se encarga de decir cómo se debe hacer un servicio para que sea realmente bueno y describe cada uno de los procesos que intervienen en un servicio.

1.3 Proceso de Gestión de Eventos

Los eventos están asociados y controlados por la Gestión de Eventos, perteneciente a la fase de Operación de Servicios según ITIL, la cual se encarga de la monitorización de los eventos que acontezcan en la infraestructura de las TI con el objetivo de garantizar su correcto funcionamiento y ayudar a prevenir futuras incidencias.

Los eventos no tienen por qué ser siempre negativos también pueden ser rutinarios, a continuación se muestran algunos ejemplos de eventos según su impacto (5):

- Eventos que indican que el servicio está operando con normalidad.
- Eventos que indican una excepción.
- Eventos que indican una operación inusual pero no excepcional, y que requieren una monitorización exhaustiva.

Algunas de las ventajas que una correcta Gestión de Eventos aporta a la organización de las TI son (5):

- Ayuda a la detección temprana de incidentes, llegando incluso a evitar que éstos se manifiesten a los usuarios.
- Además, la coordinación directa con otros procesos hace posible que éstos reaccionen con mayor rapidez, resultando en una mayor eficiencia de toda la organización TI.
- Posibilita la monitorización automatizada de determinadas actividades. Es más barata que una monitorización en tiempo real y disminuye considerablemente el periodo de inactividad del servicio que media entre la aparición del incidente y su resolución definitiva.
- Proporciona la base para las operaciones automatizadas, que incrementan la eficiencia y descargan de trabajo a los recursos humanos que, así, pueden ser empleados en otras tareas como diseñar nuevas funcionalidades.

La Gestión de Eventos (5), se encarga de detectar y notificar los sucesos, de clasificarlos y dimensionar su impacto en el servicio. Una vez que este servicio comienza su operación, es imprescindible la monitorización

y seguimiento de todos los eventos importantes que se produzcan, para poder resolverlos y prevenirlos. Como se muestra en la Ilustración 2, este servicio comienza en sí cuando aparece un evento; este se notifica mediante herramientas de gestión que analizan periódicamente los CI y el propio CI genera un reporte.



Ilustración 2. Flujo de Gestión de Evento

En la detección y filtrado, se lee el reporte generado anteriormente, se interpreta los datos relacionados con el evento y se decide si el mismo se comunica o no con la herramienta de gestión. Seguidamente se clasifica en informativo, alerta o excepción. Luego se muestra la correlación de este evento con otros, interpretándolo según la categorización y el nivel de prioridad, busca la existencia de otros elementos similares, acciones asociadas al evento y se pregunta si representa una excepción. Luego que se ha reconocido la importancia de un evento mediante la correlación, es necesario que se desencadene una tarea o varias de ellas usando los disparadores. Existen varios disparadores, los de incidentes y de cambios, los procedentes de una petición de cambio (RFC) aprobada o rechazada, entre otros. Se le da respuesta al evento; las más comunes son, registro de eventos, respuesta automática o alerta por intervención humana y emisión de una RFC.

Para dar por terminado este proceso, es necesario revisar todos los eventos importantes o excepciones, para tener la seguridad que estos han sido tratados de forma correcta. Una vez terminadas todas estas acciones, se da por cerrado el evento.

1.4 Sistemas Homólogos.

Mediante un análisis del estado del arte se estipuló que en la actualidad existen sistemas de Gestión de Eventos que siguen un flujo de trabajo definido por algún conjunto de mejores prácticas, para de cierta forma se facilite su desarrollo. Para realizar esta investigación es necesario enfocarse en varios parámetros que facilitarán la misma, siendo el de mayor importancia el uso de ITIL como conjunto de buenas prácticas. A continuación se detallan algunas de las características fundamentales de las soluciones analizadas.

Manage Engine Service Plus

Manage Engine Service Plus (13) es una solución que tiene función de Escritorio de Servicio (Service Desk). Está basado en las mejores prácticas de ITIL, ofreciendo un paquete integrado con la Gestión de Incidencias, Inventario Automático, Gestión de Activos, Gestión de Contratos y Gestión de Conocimientos. Incluye todo lo que necesita para gestionar sus procesos de informática interna, lo cual redundará en mayor productividad y mejor servicio para usuarios finales. Es fácil de implantar y su interfaz web es muy intuitivo.

Trellis Desk

Trellis Desk (14) es una solución basada en ITIL, una aplicación web de gestión del conocimiento y soporte técnico que posibilita optimizar el servicio postventa a los clientes permitiendo la gestión inteligente de solicitudes de ayuda.

Kayako

Kayako (15) es una solución Service Desk basada en ITIL, desarrollando dentro de la fase de Operación de Servicio, la Gestión de Cambios, Gestión de Problemas y Gestión de Incidencias. Ayuda a mejorar la calidad de los servicios de las TI en la empresa que lo ejecuta y eleva el nivel de satisfacción de sus usuarios.

Cerberus

Cerberus (16) es una plataforma web fácil y flexible para la colaboración empresarial, basada en ITIL y se enfoca en la Gestión de Eventos y la Gestión de Incidencias. Además cuenta con asistentes virtuales los cuales proporcionan facilidades para configurar la automatización del negocio.

CA

CA Technologies (17) es un sistema que abarca tres fases de ITIL: Diseño de Servicio, Transición de Servicio y Operación de Servicio. Dentro de la Operación de Servicio desarrolla la Gestión de Eventos, la Gestión de Incidencias, la Gestión de Conocimiento y la Gestión de Problemas. Este sistema constituye una ayuda a nuestros clientes para agilizar y simplificar las TI para aprovechar mejor las nuevas tecnologías. Reduciendo la complejidad ayuda a las empresas a centrarse en la calidad del servicio en vez de la infraestructura, lo que incluye asegurar la infraestructura de TI para reducir riesgos y mejorar la fiabilidad. Este sistema es compatible con entornos de Windows, UNIX, Linux y NetWare.

Análisis Crítico

Estos sistemas analizados no son los únicos existentes en el mundo, pero se desarrolló una investigación respecto a estos ya que poseen documentación disponible en la red para ser consultada y además usan ITIL como mejores prácticas. Este estudio nos permitió entender cómo funciona el proceso de Gestión de Eventos dentro de una empresa. Todos estos *software* no son usados como solución puesto que la mayoría son privativos y no se conocen con exactitud los procedimientos utilizados para su funcionamiento, lo que trae como consecuencia el no poder agregarle nuevas funcionalidades de importancia en el flujo de trabajo de la entidad. Además no abarcan todas las necesidades del Centro de Soporte planteadas en la situación problemática. Estos sistemas no comprenden la Gestión de Eventos como su principal funcionalidad. Es necesario aclarar que a nivel nacional no se encontró ningún sistema de Gestión de Eventos. Por lo antes expuesto se hace necesaria la implementación de un módulo que gestione los eventos que acontezcan en el Centro de Soporte y de tal forma satisfaga sus necesidades.

1.5 Herramientas y tecnologías.

Conociendo las funcionalidades y propiedades de la Gestión de Eventos, se hace necesario utilizar un conjunto de herramientas que faciliten la implementación del módulo, consumiendo la menor cantidad de recursos y de la forma más rápida posible. Las mismas fueron definidas por el cliente debido a la futura integración del módulo al Sistema de Gestión de Servicios del Centro de Soporte UCI y lograr la homogeneidad con otros módulos pertenecientes a este sistema informático. A continuación se presentan la metodología, tecnologías y herramientas a utilizar para el desarrollo del módulo de Gestión de Eventos.

1.5.1 Metodología de Desarrollo XP.

Extreme Programming (XP) (18) es una metodología ágil. Se caracteriza por el uso de técnicas para agilizar el desarrollo del software, así como una mayor flexibilidad para adaptarse a los cambios en los requisitos del

proyecto. Presenta como una de las ventajas la optimización del tiempo de desarrollo; permite realizar el desarrollo del sistema mediante la técnica de programación en parejas para complementar los conocimientos. El código es sencillo y entendible, además de la poca documentación a elaborar para el desarrollo del sistema que por lo general es usada para pequeños proyectos que contengan un número reducido de desarrolladores.

Las potencialidades que brinda esta metodología se ajustan a la solución que se va a proponer, dado al corto tiempo de desarrollo, al trabajo en pareja y proporcionar la interacción continua y directa con el cliente. Además le facilita el trabajo al equipo de desarrollo, debido a la poca generación de documentos que define la metodología.

1.5.2 Patrón Arquitectónico: MVC

Modelo-Vista-Controlador (MVC) (19) es un patrón de arquitectura de software que divide la aplicación en tres módulos con funcionalidades bien definidas: el Modelo, las Vistas y el Controlador. Las ideas centrales detrás de MVC son reutilización del código y la separación de responsabilidades.

Este patrón de arquitectura es usado en general para aplicaciones que trabajan con gran cantidad de información y una extensa base de datos. También facilita la programación en las diferentes capas de manera paralela e independiente. Como una de sus ventajas se puede mencionar, que el programador no debe preocuparse porque las vistas se actualicen, pues este proceso lo realiza de forma automática el modelo de la aplicación, además, las modificaciones de las vistas no afectan a los demás módulos.

Este patrón al tener una clara separación entre la vista, el modelo y el controlador facilita la realización de pruebas unitarias a los componentes del módulo, así como la simplicidad para el mantenimiento posterior.

1.5.3 Herramienta CASE: Visual Paradigm 8.0

Visual Paradigm (20) es una herramienta para el desarrollo de aplicaciones utilizando modelado UML, ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de Sistemas. Permite especificar y describir los métodos y procesos relacionados con el negocio.

Esta herramienta es un apoyo básico para la elaboración del modelo conceptual del módulo Gestión de Eventos.

1.5.4 Framework: Grails 2.1

Grails (21) es un *framework*⁶ de aplicaciones web dinámica basada en Java y Groovy. Este sigue la arquitectura Modelo-Vista-Controlador (MVC) que es la que se empleará en el desarrollo de la aplicación, además, se utiliza el lenguaje Groovy que permite realizar una gran cantidad de acciones en pocas líneas de código. Tiene como ventaja que toda su funcionalidad se encuentra disponible en dicho lenguaje, pues los desarrolladores de Grails eligieron a Groovy como su lenguaje base debido a su poder y dinamismo.

Este *framework* incluye una serie de *plugins*⁷ que facilitan el trabajo del desarrollador. Además sigue el patrón MVC, este definido para el desarrollo del módulo y permite utilizar las propias librerías de Java.

1.5.5 Lenguaje de Programación: Groovy 1.8

Groovy (22) es un lenguaje de programación orientado a objetos implementado sobre la plataforma Java. Usa una sintaxis muy parecida a Java, comparte el mismo modelo de objetos, de hilos y de seguridad. Desde Groovy se puede acceder directamente a todas las Interfaces de Programación de Aplicaciones (API) existentes en Java.

Este lenguaje permite realizar una gran cantidad de acciones en pocas líneas de código, además el *framework* Grails lo define como su lenguaje base.

1.5.6 IDE de Desarrollo: IntelliJ IDEA 11.0

IntelliJ IDEA (23) es un entorno de desarrollo Java, soporta varias tecnologías como: Groovy, Android, Java Script y otros. Las principales ventajas son el autocompletado de código, integración con sistemas de control de versiones y la contención de un amplio set de plugins, además presenta potentes herramientas para la gestión de bases de datos.

Este IDE de desarrollo es utilizado para el desarrollo de aplicaciones web. Facilita el trabajo directo a la base de datos permitiendo la edición de las anotaciones de esta y las solicitudes directas dentro del mismo IDE.

⁶ Marco de trabajo.

⁷ Programa que puede anexarse a otro para aumentar sus funcionalidades (generalmente sin afectar otras funciones ni afectar la aplicación principal).

1.5.7 Sistema de Gestión de Bases de Datos: PostgreSQL9.1

Postgre SQL (24) es un sistema de base de datos objeto-relacional, permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Adicionalmente los usuarios pueden crear sus propios tipos de datos.

Empleando esta herramienta se garantiza la seguridad y confiabilidad de la base de datos, mediante la gestión de usuarios y contraseñas. Además es uno de los gestores más recomendados para sistemas que guarden gran cúmulo de información.

1.6 Conclusiones

El análisis del estudio sobre sistemas homólogos a nivel internacional de la Gestión de Eventos, determinó que ninguno de los encontrados satisface las especificaciones del cliente lo que comprobó la necesidad de desarrollar un módulo que gestione los eventos que acontezcan en el Centro de Soporte UCI. Se definió la metodología de desarrollo, las herramientas y tecnologías a utilizar en el desarrollo de la solución propuesta, las cuales permitirán una implementación eficiente en corto tiempo.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN DEL MÓDULO DE GESTIÓN DE EVENTOS.

En el presente capítulo se describe cómo se gestiona los eventos de servicios TI en el Centro de Soporte UCI. Inicialmente se muestra la solución propuesta mostrando las prestaciones que brinda la Gestión de Eventos, los roles y responsabilidades, y los requisitos funcionales y no funcionales del módulo. Se definen las HU⁸ con su estructura y algún ejemplo de ellas, el plan de iteraciones y se muestra el estudio realizado de los patrones de diseño y los estándares de codificación utilizados en el desarrollo del módulo. Se explica el proceso de Gestión de Eventos de forma breve por medio del modelo conceptual, se describen las tarjetas CRC⁹ y se muestra el diseño de la base de datos.

2.1 Modelo Conceptual

Mediante el modelo conceptual se proporciona una vista estructural del sistema actual. Se describen las principales entidades implicadas en el proceso para llevar el registro de eventos en el Centro de Soporte UCI.

Actualmente es muy engorroso todo este proceso de llevar a cabo una correcta Gestión de Eventos de los servicios en el centro, debido a que el sistema con el que consta, no satisface estas necesidades específicas. Muchas veces los clientes no están de acuerdo con los tiempos de respuesta de los servicios prestados, lo que trae consigo una insatisfacción de los mismos. A continuación se muestra el modelo conceptual.

⁸ Historia de Usuarios.

⁹ Clases, Responsabilidades y Colaboraciones.



Ilustración 3. Modelo Conceptual

Cuando un proceso determinado provoca la ocurrencia de un evento, este es identificado por un especialista y luego controlado por el director del Centro de Soporte. Este proceso se realiza de forma manual, ya que el sistema con el que se cuenta actualmente no tiene integrado la Gestión de Eventos.

2.2 Solución Propuesta

El módulo de Gestión de Eventos está concebido como una aplicación web, a través de la cual se registrarán todos los eventos que acontezcan en el sistema. El objetivo del módulo es prevenir posibles afectaciones en el futuro e informar a los especialistas sobre el impacto de estos eventos en los distintos procesos de la Gestión de Servicios. Este deberá ser actualizado con toda la información generada durante el proceso de eventos para permitir un detallado seguimiento desde su aparición hasta su cierre.

La Gestión de Eventos representa una tarea en sí misma y constituye un proceso independiente dentro del ciclo de vida que además de detectar y notificar los sucesos, se encarga de clasificarlos y dimensionar su impacto en el servicio. Llegado el caso, se ocupa también de documentar el evento y derivarlo al proceso correspondiente para que tome medidas:

- A la Gestión de Incidencias, en caso de que el evento suponga una interrupción no planificada del servicio o fallos en uno o más elementos de configuración.
- A la Gestión de Problemas, si una incidencia se repite a menudo y no se conoce la causa que la provoca.
- Y también envía a la Gestión de Cambios, nuevas solicitudes de cambio basadas en la correlación de eventos.

El ciclo de vida del evento comienza con su aparición, se le realiza la notificación, seguidamente se clasifica según categorías, se le otorga un impacto para llegar a la solución y se le realiza el cierre. Para tener un

mejor entendimiento de lo antes expuesto se describen, de forma detallada, los datos que se manejan en cada una de las actividades que posee un evento:

- La notificación se le realiza a los usuarios vinculados al evento.
- La clasificación del evento se realiza según las categorías (Alerta, Informativo, Excepción) y las sub_categorías. Estas serán definidas con la posterior integración del módulo al Sistema de Gestión de Servicios.
- El impacto se otorga atendiendo al grado de afectación a otros procesos (Incidencias, Problemas, Acceso y Cumplimientos de Solicitudes).
- El cierre se efectúa cuando se ejecuten correctamente todas las fases.

Conjuntamente el módulo reúne entre sus amplias prestaciones la muestra de gráficas estadísticas facilitando la toma de decisiones y cuenta con una base de datos de los eventos generados en el sistema.

2.3 Roles y responsabilidades

Roles	Responsabilidades
Administrador	Responsable de mantener la disponibilidad de los datos y la correcta gestión del proceso, además de darle cierre a un evento y de administrar en el módulo, los niveles de acceso de los usuarios.
Supervisor	Responsable de declarar un documento final, además que presenta las mismas responsabilidades del Jefe de Grupo.
Jefe de Grupo	Responsable de la actualización del historial de eventos, de todas las modificaciones que se lleven a cabo a los documentos de eventos que formen parte del equipo de trabajo y responsable de generar reportes y notificaciones de eventos.
Usuario	Cuenta con las responsabilidades que le otorga el administrador.

Tabla 1. Roles y responsabilidades

2.4 Requisitos Del Sistema

2.4.1 Requisitos Funcionales

Definir requisitos funcionales en un sistema es primordial, pues estos describen los servicios proporcionados por el sistema y las restricciones operativas. Es la condición o la capacidad que tiene que ser alcanzada por un sistema para satisfacer las necesidades de los clientes, solucionando alguna deficiencia o problema con ayuda de un producto informático.

A continuación se definen los requisitos funcionales del sistema:

RF 1: Autenticar usuario	RF 5.1: Crear categoría
RF 2: Gestionar usuario	RF 5.2: Modificar categoría
RF 2.1: Crear usuario	RF 5.3: Eliminar categoría
RF 2.2: Modificar usuario	RF 5.4: Buscar categoría
RF 2.3: Eliminar usuario	RF 6: Gestionar sub_categoría
RF 2.4: Buscar usuario	RF 6.1: Crear sub_categoría
RF 3: Gestionar grupo de usuarios	RF 6.2: Modificar sub_categoría
RF 3.1: Crear grupo de usuarios	RF 6.3: Eliminar sub_categoría
RF 3.2: Modificar grupo de usuarios	RF 6.4: Buscar sub_categoría
RF 3.3 Eliminar grupo de usuarios	RF 7: Visualizar impacto del evento
RF 3.4: Buscar grupo de usuarios	RF 8: Visualizar Procesos Afectados
RF 4: Gestionar evento	RF 9: Mostrar Matriz de Eventos
RF 4.1: Crear evento	RF 10: Realizar notificaciones de los eventos a usuarios.
RF 4.2: Modificar evento	RF 11: Exportar el listado de eventos en formato pdf.
RF 4.3: Eliminar evento	RF 12: Listar procesos afectados por un evento.
RF 4.4: Buscar evento	RF 13: Realizar búsquedas por parámetros.
RF 5: Gestionar categoría	RF 14: Visualizar gráficas estadísticas de los eventos.

Tabla 2. Requisitos Funcionales del módulo

2.4.2 Requisitos No Funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Estos a menudo se aplican al sistema en su totalidad (25).

Usabilidad

RNF 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.

Diseño e implementación

RNF 2 Se utilizará Postgre SQL en su versión 9.1 o superior como Sistema Gestor de Bases de Datos.

RNF 3 Se utilizará Groovy como Lenguaje de Programación y Grails como Framework Integrador.

RNF 4 Se utilizará como arquitectura del sistema Modelo- Vista- Controlador.

RNF 5 Se utilizará IntelliJ Idea como IDE para el desarrollo del sistema.

RNF 6 Se utilizará Windows 7 o superior como Sistema Operativo para el desarrollo del sistema.

RNF 7 Se utilizará Visual Paradigm como herramienta de modelado.

RNF 8 Para el diseño de las páginas se utilizarán las CSS¹⁰, HTML¹¹, XHTML¹², Java Script, JQuery.

Funcionamiento

Software:

RNF 9 Para el funcionamiento de los servidores es necesario Windows 7 o superior.

RNF 10 Para el funcionamiento en la PC cliente será necesario Firefox 6 o superior, Google Chrome 16 o superior. Sistema operativo Windows o Mac capaces de soportar los navegadores mencionados anteriormente.

Hardware:

RNF 11 El servidor para aplicaciones web deberá tener las siguientes características mínimas:

¹⁰ Hojas de estilo en cascadas.

¹¹ Lenguaje de marcas hipertexto.

¹² Lenguaje de marcas hipertexto extensible.

- 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 100 GB de disco duro.

RNF 12 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.

RNF 13 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:

RNF 14 El acceso al sistema así como la información se encontrarán protegidos contra accesos no autorizados utilizando mecanismos de autenticación propios del sistema.

RNF 15 La autenticación del sistema será la primera acción del usuario, el cual deberá proporcionar un usuario único y una contraseña, los cuales serán de uso exclusivo del propio usuario.

RNF 16 Las diferentes áreas del sistema se encontrarán protegidas contra acceso no autorizado utilizando roles y grupos de usuarios.

RNF 17 El administrador del sistema como política de seguridad podrá restringir el acceso a las diferentes áreas del sistema a los usuarios o grupos de usuarios.

Integridad:

RNF 18 La información podrá ser modificada solo por el personal autorizado.

RNF 19 Se harán validaciones de la información en el servidor contra ataques de inyección HTML¹³ y SQL¹⁴.

¹³ Lenguaje de marcas de hipertexto.

RNF 20 Se protegerá la información almacenada contra ataques CSRF¹⁵.

Fiabilidad

RNF 21 La respuesta del sistema ante una búsqueda bajo criterios no debe exceder los 15 segundos.

RNF 22 Ante una inserción de datos, vista de detalles, modificación o eliminación el sistema debe responder en no más de 5 segundos.

RNF 23 El sistema debe responder en un promedio de 10 segundos la petición de un reporte.

RNF 24 Ninguna información ingresada en el sistema será eliminada físicamente de la base de datos.

Eficiencia

RNF 25 Se respetarán buenas prácticas de programación para incrementar el rendimiento en operaciones costosas.

Interfaz de usuario

RNF 26 Las interfaces del sistema contendrán los datos de forma estructurada, permitiendo la interpretación correcta de la información.

RNF 27 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.

RNF 28 Todos los textos y mensajes en pantalla serán mostrados según el idioma seleccionado para el sistema.

RNF 29 El diseño de la interfaz del sistema responderá a la ejecución de acciones de forma rápida, minimizando los pasos a dar en cada proceso.

Interconexión

¹⁴ Lenguaje de consulta estructurado.

¹⁵ Falsificación de petición en sitios cruzados.

RNF 30 La PC cliente se comunicará mediante el HTTP¹⁶ y HTTPS¹⁷ con el servidor de aplicaciones, éste se comunicará mediante el TCP/IP¹⁸ con el servidor de bases de datos.

RNF 31 Los datos de usuarios y demás datos necesarios para el sistema podrán ser extraídos de los servicios web brindados por el UDDI (Directorio de Servicios web de la UCI).

2.5 Historias de Usuario

Las HU tienen el mismo propósito que los casos de uso, pero tiene sus diferencias. Las escriben los propios clientes, tal y como observan ellos las necesidades del sistema. Por tanto, serán descripciones cortas y escritas en el lenguaje del usuario, sin terminología técnica (26).

Las Historias de usuarios cuentan con la estructura que se explica a continuación:

Número: A cada HU se le asigna un número para facilitar su identificación por parte del equipo de desarrollo.

Nombre: Nombre descriptivo de la HU.

Prioridad en Negocio: Grado de prioridad que le asigna el cliente a la HU en dependencia de sus necesidades. Los valores que puede tomar son (Alta, Media o Baja).

Riesgo en Desarrollo: Grado de complejidad que le asigna el equipo de desarrollo a la HU luego de analizarla. (Alto, Medio o Bajo).

Puntos Estimados: Unidades de tiempo estimadas por el equipo de desarrollo para darle cumplimiento a la HU. Una unidad de tiempo equivale a una semana de trabajo de 40 horas, por lo que un día de trabajo se representaría por 0,2 unidades que sería el equivalente a 8 horas laborales.

Puntos Reales: Unidades de tiempo reales que el equipo de desarrollo necesitó para darle cumplimiento a la HU. Una unidad de tiempo equivale a una semana de trabajo de 40 horas, por lo que un día de trabajo se representaría por 0,2 unidades que sería el equivalente a 8 horas laborales.

Iteración Asignada: Número de la iteración en la cual será implementada la HU.

¹⁶ Protocolo de transferencia de hipertexto.

¹⁷ Protocolo de transferencia de hipertexto seguro.

¹⁸ Protocolo de control de transmisión/Protocolo de Internet.

Descripción: Descripción simple sobre lo que debe hacer la funcionalidad a la que se hace referencia.

Observaciones: Condiciones que deben tenerse en cuenta para el desarrollo de la funcionalidad.

A continuación se muestran unas de las historias de usuario más relevantes y las restantes se pueden visualizar en el [Anexo 1](#).

Historia de Usuario	
Número: 1	Nombre de la Historia de Usuario: Autenticar usuario
Cantidad de modificaciones a la Historia de Usuario: Ninguna	
Usuario: Usuario	Iteración asignada: 1
Prioridad en negocio: Muy Alta	Puntos estimados: 2 y ½ semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
<p>Descripción:</p> <ol style="list-style-type: none"> 1. El sistema muestra la interfaz principal. 2. El usuario introduce sus datos personales con su contraseña. 	
<p>Observaciones: Si no coincide la contraseña con el usuario la aplicación mostrará un mensaje informando que la contraseña es incorrecta.</p>	



Tabla 3. HU Autenticar Usuario.

Historia de Usuario	
Número: 5	Nombre de la Historia de Usuario: Gestionar evento.
Cantidad de modificaciones a la Historia de Usuario: Ninguna	
Usuario: Supervisor	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Descripción: <ol style="list-style-type: none"> 1. El sistema muestra la interfaz correspondiente a la gestión de evento. 2. El administrador selecciona la opción "Adicionar Evento". 3. El sistema muestra un conjunto de parámetros correspondiente al evento. 	

4. El administrador completa los datos.
5. El sistema muestra un cuadro de diálogo, informándole al administrador que el evento ha sido creado correctamente.
6. El administrador tiene la posibilidad de modificar el evento añadido, seleccionándolo y modificando los campos necesarios, además de eliminarlo en la opción de Eliminar Evento.

Observaciones: Si los datos de creación del evento no son válidos la aplicación mostrará mensajes con los errores correspondientes.

Prototipo de interfaces:

Criterios de búsqueda

Nombre:

Subcategoría:

Categoría:

Urgencia:

Visualizar matriz
Exportar
Graficar
+ Adicionar

Listado de eventos

NOMBRE	CATEGORÍA	SUBCATEGORÍA	URGENCIA	
Evento	Alerta	Alerta de Mantenimiento	Sí	
Evento1	Alerta	Alerta de Mantenimiento	Sí	

Tabla 4. HU Gestionar Evento.

Número: 2	Nombre de la Historia de Usuario: Gestionar usuario
Cantidad de modificaciones a la Historia de Usuario: Ninguna	
Usuario: Administrador	Iteración asignada: 1
Prioridad en negocio: Muy Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
<p>Descripción:</p> <ol style="list-style-type: none"> 1. El sistema muestra la interfaz principal. 2. El administrador introduce sus datos con su contraseña. 3. El sistema muestra una interfaz según sus permisos administrativos. 4. El administrador pulsa en la opción "Adicionar". 5. El sistema muestra un conjunto de parámetros correspondientes al usuario. 6. El administrador completa los datos. 7. El sistema guarda todos los datos referentes al usuario adicionado al sistema. 8. El administrador tiene la posibilidad de modificar el usuario añadido, seleccionándolo y modificando los campos necesarios, además de eliminarlo. 	
<p>Observaciones: Si los datos de creación del usuario no son válidos la aplicación mostrará mensajes con los errores correspondientes.</p>	

Prototipo de interfaces:

Criterios de búsqueda

Nombre: Correo:

Rol: CI:

 Eliminar  Adicionar

Listado de usuarios

<input type="checkbox"/>	NOMBRE	CORREO ELECTRÓNICO	GRUPO	CI	HABILITADO	
<input checked="" type="checkbox"/>	Tatyana	tmiranda@estudiantes.uci.cu	Administrador	91012644251	Sí	

Adicionar usuario

DATOS DEL NUEVO USUARIO

Nombre completo

Correo electrónico

Grupo de usuario

Carné de Identidad

Contraseña

The image shows a web interface for user management. The top section is a form titled 'Modificar usuario' with the subtitle 'DATOS DEL USUARIO A MODIFICAR'. It contains several input fields: 'Nombre completo' (Tatyana), 'Correo electrónico' (tmiranda@estudiantes.uci.cu), 'Grupo de usuario' (Administrador), 'Carnet Identidad' (91012644251), and 'Contraseña' (with a 'Repetir Contraseña' field). There are 'Modificar' and 'Cancelar' buttons. Below the form is a table titled 'Lista Usuarios' with columns: NOMBRE, CORREO ELECTRÓNICO, GRUPO, CI, and HABILITADO. The table contains one row for 'Tatyana' with email 'tmiranda@estudiantes.uci.cu', group 'Administrador', CI '91012644251', and status 'Sí'. There are 'Eliminar' and 'Adicionar' buttons above the table.

Tabla 5. HU Gestionar Usuario.

2.6 Plan de Iteraciones

En la metodología XP, la creación del sistema se divide en etapas para facilitar su realización. Por lo general, los proyectos constan de 3 iteraciones. Para cada iteración se define un módulo o conjunto de historias que se van a implementar. Al final de cada iteración se obtiene como resultado la entrega del módulo correspondiente, el cual debe haber superado las pruebas de aceptación que establece el cliente para verificar el cumplimiento de los requisitos. Las tareas que no se realicen en una iteración se tienen en cuenta para la próxima iteración, donde se define, junto al cliente, si se deben realizar o si deben ser removidas de la planeación del sistema (27).

Historias de Usuarios	Duración Estimada	Iteración
Autenticar usuario.	8 semanas	1

Gestionar usuario.		
Gestionar grupo de usuarios.		
Gestionar evento.		
Gestionar categoría.		
Gestionar sub-categoría.	9 semanas	2
Visualizar impacto del evento		
Visualizar Procesos Afectados		
Mostrar Matriz de Eventos		
Permitir realizar notificaciones de los eventos a usuarios.	6 semanas	3
Permitir exportar el listado de eventos en diferentes formatos pdf.		
Listar procesos afectados por un evento.		
Realizar búsquedas por parámetros.		
Permitir visualizar gráficas estadísticas de los eventos		

Tabla 6. Plan de Iteraciones

2.7 Tarjeta CRC

Las tarjetas CRC significan las relaciones que existen entre Clases, Responsabilidades y Colaboraciones, permitiendo al equipo de desarrollo entender la posición de cada objeto dentro del espectro del software. Las tarjetas CRC permiten desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetos. (28).

A continuación se muestran unas de las Tarjetas CRC más relevantes y las restantes se pueden visualizar en [Anexo 2](#).

Tarjeta CRC
Clase: EventoController

Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> • Es la encargada de: <ul style="list-style-type: none"> ○ addProceso() ○ addUsuario() ○ removeUsuario() ○ removeProceso() ○ Notify() ○ countNotif() ○ exportar() ○ FileOutputStreamGeneratePdf(text, title, imageName) ○ addTitlePage(Document document, title, String imageName) • Además de las funcionalidades que genera de forma automática el framework: <ul style="list-style-type: none"> ○ list(Integermax) ○ create() ○ save() ○ show(Long id) ○ edit(Long id) ○ update(Long id, Long version) ○ delete(Long id) 	<ul style="list-style-type: none"> ○ Evento ○ Categoría ○ Usuarios ○ Documentos ○ Sub_categoría ○ Urgencia ○ Notificación ○ Impacto ○ Grupo_Usuario ○ Departamento

Tabla 7. Tarjeta CRC#2 EventoController

Tarjeta CRC	
Clase: UsuarioController	
Responsabilidades	Colaboraciones

<ul style="list-style-type: none"> • Es la encargada de: <ul style="list-style-type: none"> ○ userCanBeDeleted (Usuario user) ○ whyCanBeDeleted (Usuario user) ○ index() ○ validarCorreo() • Además de las funcionalidades que genera de forma automática el framework: <ul style="list-style-type: none"> ○ list(Integermax) ○ create() ○ save() ○ show(Long id) ○ edit(Long id) ○ update(Long id, Long version) ○ delete(Long id) 	<ul style="list-style-type: none"> ○ Usuario
--	---

Tabla 8. Tarjeta CRC#6 UsuarioController.

2.8 Patrones de Diseño

2.7.1 Patrón de diseño

Un patrón de diseño es (29):

- Una solución estándar para un problema común de programación.
- Una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios.
- Un proyecto o estructura de implementación que logra una finalidad determinada.
- Un lenguaje de programación de alto nivel.
- Una manera más práctica de describir ciertos aspectos de la organización de un programa.
- Conexiones entre componentes de programas.
- La forma de un diagrama de objeto o de un modelo de objeto.

Patrones GRASP

Los patrones GRASP¹⁹ realizan descripción de los principios fundamentales de la asignación de responsabilidades expresados como patrones. La asignación de responsabilidades es la habilidad más importante en el análisis y diseño orientado por objetos (30).

En el diseño del módulo se hará uso de los patrones de diseño tales como: Experto, Creador, Controlador, Alta Cohesión y Bajo Acoplamiento. A continuación se dará una explicación breve de cada uno ejemplificándolo con su uso en la implementación y diseño del sistema.

Patrón Experto

Es el principio básico de asignación de responsabilidades en diseño orientado a objetos. Se encarga de asignar una responsabilidad al experto en información, o sea, aquella clase que cuenta con la información necesaria para cumplir la responsabilidad. En el módulo se pone de manifiesto a través de la siguiente ilustración, que muestra como la clase EventoController delega la responsabilidad de exportar a la clase Evento, ya que esta reúne la información necesaria para realizar dicha funcionalidad, lo que la hace ser, la clase experta en la información:



Ilustración 4. Ejemplo del patrón experto

Patrón Creador

¹⁹ Patrones Generales de Software para Asignar Responsabilidades.

Guía la asignación de responsabilidades relacionadas con la creación de objetos. En el sistema se evidencia mediante la clase EventoController, ya que tiene la responsabilidad de crear una instancia de la clase Evento, o sea que EventoController es un creador de los objetos de Evento, como se muestra a través de la siguiente ilustración:



Ilustración 5. Ejemplo del patrón creador

Patrón Controlador

El patrón controlador se evidencia en el módulo mediante las clases controladoras, que son generadas a su vez por el patrón arquitectónico usado, más específicamente se pone de manifiesto como se muestra a través de la siguiente ilustración con la clase EventoController, que es la intermediaria entre la vista Evento y el algoritmo que la implementa, ya que recibe los datos del usuario y los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control.

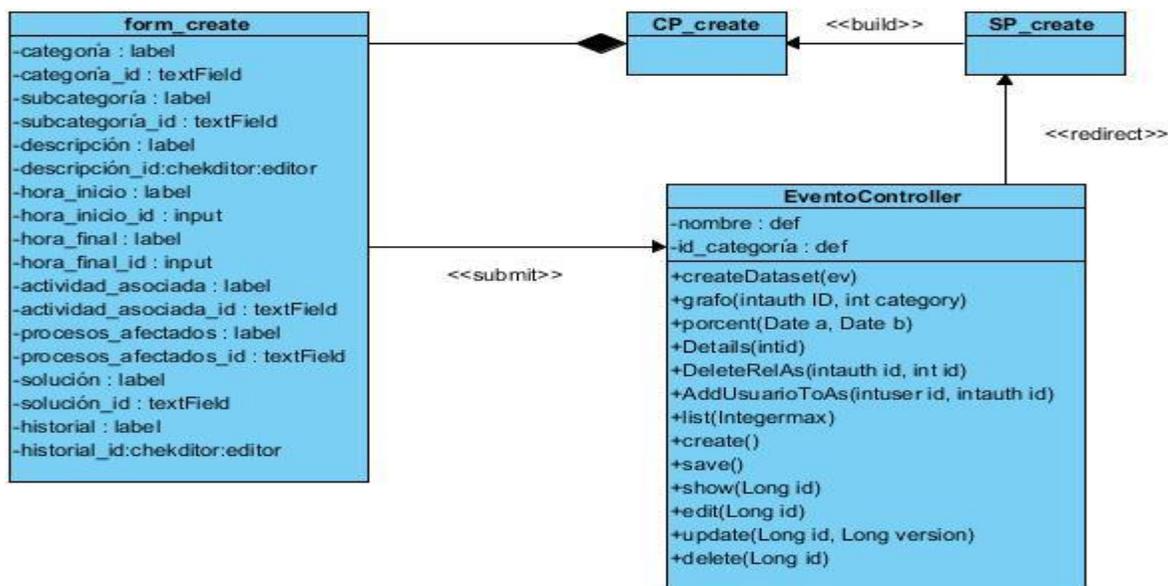


Ilustración 6. Ejemplo del patrón controlador

Patrón Alta Cohesión

La información que almacena una clase debe de ser coherente y está en la mayor medida de lo posible relacionada con la clase. La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una clase con alta cohesión, compartirá la responsabilidad de una operación, con otras clases. Se evidencia en la clase Evento, puesto que está relacionada con una serie de clases de la cuales requiere para crear una instancia de ella como por ejemplo, la clase Categoría.

Patrón Bajo Acoplamiento

Es la idea de tener las clases lo menos ligadas entre sí. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases. Los beneficios de este patrón es que no se afectan por cambios de otros componentes, son fáciles de entender por separado y fáciles de reutilizar.

2.8.2 Patrón de Arquitectura

Patrón Modelo-Vista-Controlador (MVC)

El patrón MVC separa el modelado del dominio, la presentación y las acciones basados en las entradas del usuario en tres clases apartes. El modelo maneja el comportamiento y los datos del dominio de la aplicación, responde a los requerimientos de información acerca de su estado (usualmente desde la vista) y responde a las instrucciones para cambiar de estado (usualmente desde el controlador). La vista maneja el despliegue

de la información. El controlador interpreta las acciones del usuario de teclado y ratón, informando al modelo y/o a la vista para cambiar apropiadamente sus estados (31). Tanto la vista como el controlador dependen del modelo. Sin embargo, el modelo no depende ni de la vista ni del controlador como se muestra en la siguiente ilustración:

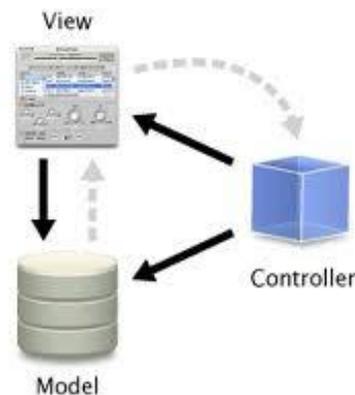


Ilustración 7. Estructura del patrón de arquitectura MVC

Modelo: Es la representación de la información que maneja la aplicación, los datos puros, que puestos en contextos del sistema, proveen de información al usuario o a la aplicación en sí. A continuación se muestra la estructura de la clase Evento como ejemplo del modelo existente en el sistema.

```

Evento.groovy x
Evento EventoController Evento Views Tests
package my

class Evento {
    int id
    String nombre
    String descripcion
    Date fecha_inicio
    Date fecha_fin
    boolean urgencia
    String historia

    static constraints = {
        // id(unique: true, nullable: false)
        // Nombre(nullable: true)
    }

    static mapping = {
        table('evento')
        version(false)
        id column:'id' //generator:'identity',
    }

    static hasMany =
        [
            procesosAfectados: Proceso,
            usuariosGeneran: Usuario,
            notificaciones: Notificacion
        ]
    }
    
```

Ilustración 8. Ejemplo del componente Modelo

Vista: Es la representación del modelo en forma gráfica, disponible para la interacción del usuario. En el caso de una aplicación Web, la vista es una página con contenido dinámico sobre el cual el usuario puede realizar operaciones. Se muestra a continuación el paquete que conforma la vista correspondiente a la clase Evento en el módulo.

```

C Evento.groovy x index.gsp x
html body div.panel.panel-primary div.panel-body table tr td
Evento EventoController:index Evento Views Tests
<@ page import="my.Grupo" contentType="text/html; charset=UTF-8" @>
<html>
<head>
<meta name="layout" content="main" />
<link rel="shortcut icon" href="<g:createLinkTo dir='images' file='clients.jpg'/>" />
<link rel="stylesheet" href="{resource(dir:'js',file:'/redmond/jquery-ui-1.8.2.custom.css')}"/>
<link rel="stylesheet" href="{resource(dir:'css',file:'cssshake.css')}"/>
<link rel="stylesheet" href="{resource(dir:'css',file:'animate.css')}"/>

<title>SRE - Eventos</title>
</head>
<body>
<div id="divalert" class="bb-alert alert alert-info" style="display: none;">
<span>
</span>
</div>
<g:form>
<div class="panel panel-primary">
<div class="panel-heading" style="text-align: left;">
Criterios de búsqueda
</div>
<div class="panel-body">
<table>
<tr>

```

Ilustración 9. Ejemplo del componente Vista

En la ilustración siguiente se muestra el resultado de la vista correspondiente a la clase Evento desde la web.

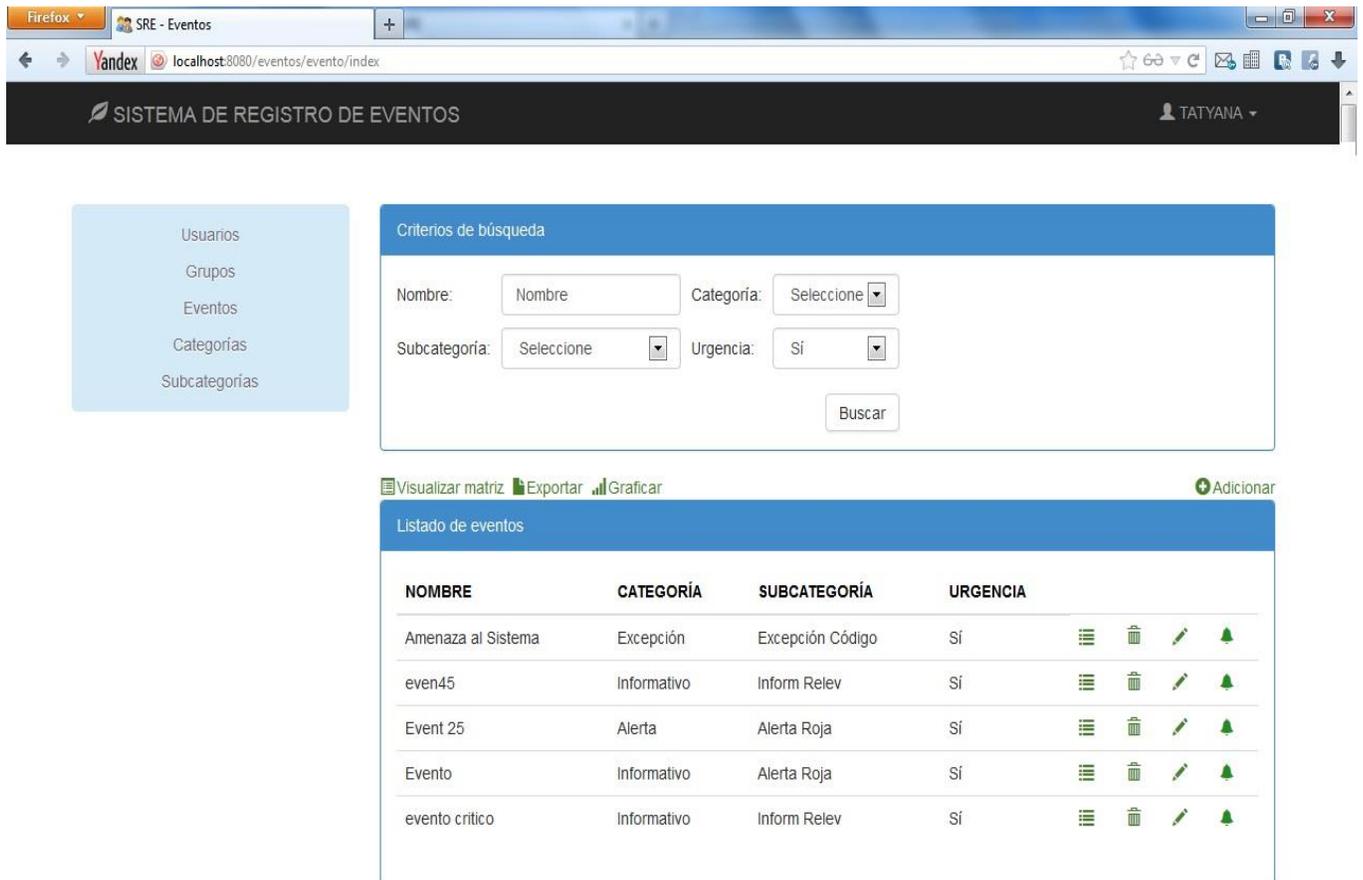


Ilustración 10. Ejemplo del componente Vista desde la Web

Controlador: Es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información necesaria modificando el Modelo en caso de ser necesario. A continuación se muestra la estructura de la clase EventoController en el sistema.

```

EventoController.groovy x
Evento EventoController Evento Views Tests
package my
import ...
class EventoController {
    static def procs = [] //guardos los ids de los procesos afectados seleccionados
    static def usuariosEvento = [] //guardos los ids de los usuarios del evento
    def plotPage={
        render view: '/grafica/index'
    }
    def index()
    {
        def offset = params.offset ?: 0
        def maxResults = params.max ?: 5
        def nombre = params.nombre
        def idcategoria = params.categoria
        def idsubcategoria = params.subcategoria
        def urgencia = params.urgencia

        if(!nombre || nombre=="")
            && (!idcategoria || idcategoria=="-1")
            && (!idsubcategoria || idsubcategoria=="-1")
            && (!urgencia || urgencia=="-1")
    }
}
    
```

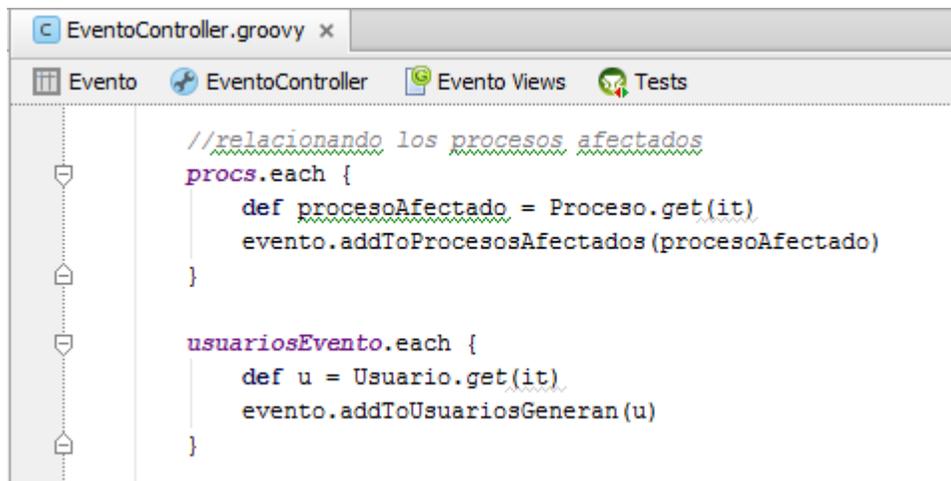
Ilustración 11. Ejemplo del componente Controlador

2.9 Diseño de Base de Datos

A continuación se presenta el modelo de Base de Datos, que de una manera lógica y estructurada muestra cómo se almacenan, organizan y manipulan los datos del sistema. En un modelo de base de datos, todos los datos se almacenan y se accede a ellos por medio de relaciones.

- **Upper Camel Case**, cuando la primera letra de cada una de las palabras es mayúscula.
 - Ejemplo: EjemploDeUpperCamelCase.
- **Lower Camel Case**, igual que la anterior con la excepción de que la primera letra es minúscula.
 - Ejemplo: ejemploDeLowerCamelCase.

En el módulo se utiliza como estándar de codificación el lowerCamelCase para nombrar los atributos y métodos de cada una de las clases existentes, a continuación se muestra el ejemplo de la estructura de la clase EventoController.



```

EventoController.groovy x
Evento EventoController Evento Views Tests

//relacionando los procesos afectados
procs.each {
    def procesoAfectado = Proceso.get(it)
    evento.addToProcesosAfectados (procesoAfectado)
}

usuariosEvento.each {
    def u = Usuario.get(it)
    evento.addToUsuariosGeneran(u)
}
    
```

Ilustración 13. Ejemplo del uso del Camel Case

2.11 Diagrama de Despliegue

Los diagramas de despliegue (35) muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Como se muestra en la Ilustración 14 el diagrama de despliegue.

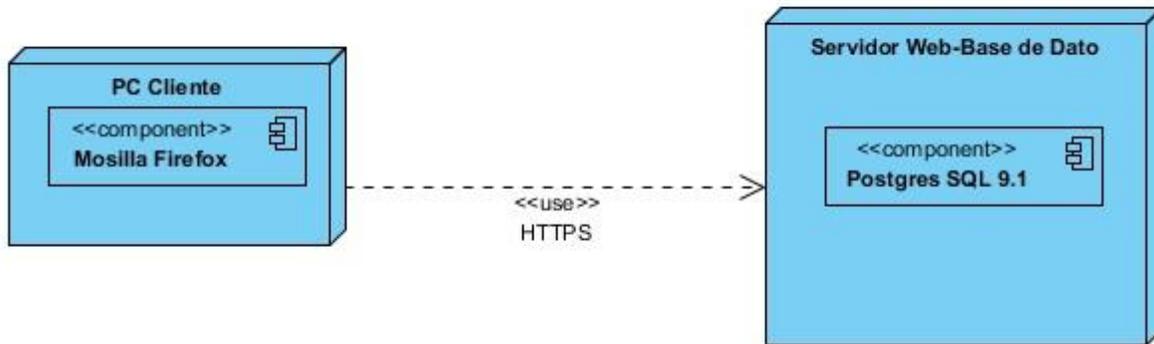


Ilustración 14. Diagrama de Despliegue del módulo Gestión de Eventos

2.12 Conclusiones

Con la elaboración de los artefactos que define la metodología XP, se garantizó una organización en el desarrollo del módulo y se logró simplificar el trabajo del implementador. Además se describieron los patrones de diseño y estándares de codificación que se utilizan durante la implementación del sistema, los cuales permiten una legibilidad del código fuente y una mayor reutilización por parte del usuario. Con la culminación de la implementación del módulo se logró dar respuesta al problema a resolver.

CAPÍTULO 3. VALIDACIÓN DEL MÓDULO DE GESTIÓN DE EVENTOS.

En el presente capítulo se analiza desde el punto de vista de la calidad, el cumplimiento del problema a resolver del trabajo de diploma, verificando la adecuada implementación de las HU definidas. Para ellos se diseñan las pruebas correspondientes a la metodología XP, las cuales se dividen en dos partes, unitarias y de aceptación. Se describe de forma detallada todo el proceso de dichas pruebas y se muestra el análisis del resultado final, que explica y muestra al cliente si el módulo está implementado correctamente y si cuenta con las funcionalidades descritas en el capítulo anterior.

3.1 Pruebas Unitarias

Uno de los tipos de pruebas que establece XP son las pruebas unitarias (33). Según la metodología son utilizadas para verificar el funcionamiento del sistema facilitándose además con la técnica de programación en parejas. *“Las pruebas unitarias son otra forma de comunicación, ya que describen el diseño de las clases y los métodos al mostrar ejemplos concretos de cómo utilizar su funcionalidad. Los programadores se comunican constantemente gracias a la programación en parejas”.*

Una prueba unitaria es la verificación de un módulo determinado dentro de un sistema. Son llevadas a cabo por los programadores de cada módulo y poseen gran beneficio ya que mediante ellas, se puede asegurar que un determinado módulo cumpla correctamente su funcionalidad.

Camino Básico: El método del camino básico es una técnica de prueba de caja blanca propuesta principalmente por Tom McCabe. Permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedural, y utilizar esta medida como guía para la definición de un conjunto básico de caminos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez (33).

Para realizar estas pruebas se utiliza la complejidad ciclomática de McCabe, que consiste en la ejecución de un conjunto de caminos independientes proporcionando una medición cuantitativa de la complejidad lógica de un programa, así como determinar el número de casos de prueba que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez. La complejidad ciclomática $V(G)$, de un grafo de flujo G consta de tres formas para calcularse y se define como:

$V(G) = R$ donde R : n° de regiones del grafo

$V(G) = A - N + 2$ donde A : n° de arcos del grafo, N : n° de nodos

$V(G) = P + 1$ donde P: nº de nodos predicado

A continuación se detalla una de las pruebas unitarias desarrolladas al sistema, en este caso la del camino básico realizadas a varias funcionalidades del módulo.

Método eventos ()

Usando el código como base, se dibuja el correspondiente grafo de flujo, como se muestra en la Ilustración 12 .

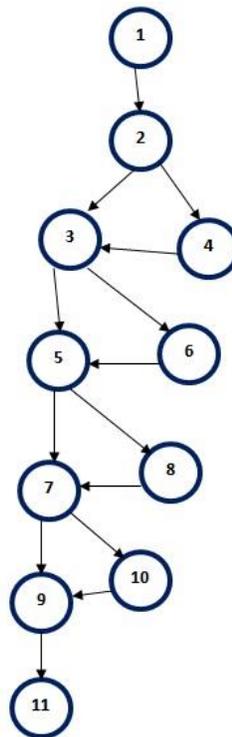


Ilustración 15. Grafo del flujo asociado al método eventos ()

Se determina la complejidad ciclomática del grafo de flujo resultante $V(G)$, la cual coincide con el número de regiones $V(G) = 5$:

$$V(G) = A - N + 2 = 14 - 11 + 2 = 5$$

$$V(G) = P + 1 = 4 + 1 = 5$$

Luego de determinada la complejidad ciclomática se tiene que 5 es el número máximo de caminos a tener en cuenta para realizar las pruebas.

Caminos independientes determinados:

Camino 1: 1-2-3-5-7-9-11

Camino 2: 1-2-3-6-5-7-9-11

Camino 3: 1-2-3-5-8-7-9-11

Camino 4: 1-2-3-5-7-10-9-11

Camino 5: 1-2-4-3-5-7-9-11

Método save ()

Usando el código como base, se dibuja el correspondiente grafo de flujo, como se muestra en la Ilustración 13 .

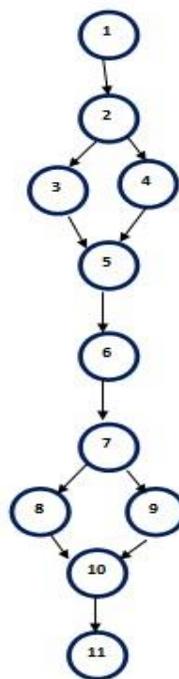


Ilustración 16. Grafo del flujo asociado al método save ()

Se determina la complejidad ciclomática del grafo de flujo resultante $V(G)$, la cual coincide con el número de regiones $V(G) = 3$:

$$V(G) = A - N + 2 = 12 - 11 + 2 = 3$$

$$V(G) = P + 1 = 2 + 1 = 3$$

Luego de determinada la complejidad ciclomática se tiene que 3 es el número máximo de caminos a tener en cuenta para realizar las pruebas.

Caminos independientes determinados:

Camino 1: 1-2-3-5-6-7-8-10-11

Camino 2: 1-2-4-5-6-7-9-10-11

Camino 3: 1-2-3-5-6-7-9-10-11

Método jsondata ()

Usando el código como base, se dibuja el correspondiente grafo de flujo, como se muestra en la ilustración.

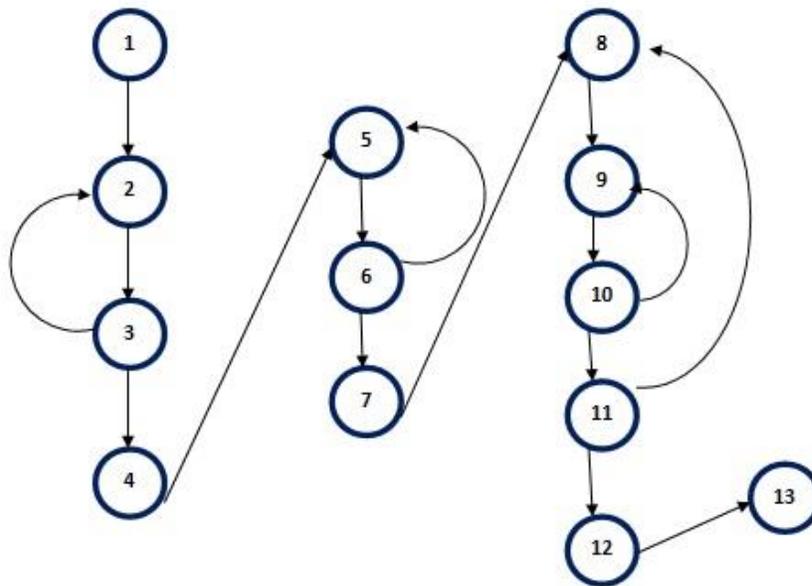


Ilustración 17. Grafo del flujo asociado al método jsondata ()

Se determina la complejidad ciclomática del grafo de flujo resultante $V(G)$, la cual coincide con el número de regiones $V(G) = 5$:

$$V(G) = A - N + 2 = 16 - 13 + 2 = 5$$

$$V(G) = P + 1 = 4 + 1 = 5$$

Luego de determinada la complejidad ciclomática se tiene que 5 es el número máximo de caminos a tener en cuenta para realizar las pruebas.

Caminos independientes determinados:

Camino 1: 1-2-3-4-5-6-7-8-9-10-11-12-13

Camino 2: 1-2-3-2-3-4-5-6-7-8-9-10-11-12-13

Camino 3: 1-2-3-4-5-6-5-6-7-8-9-10-11-12-13

Camino 4: 1-2-3-4-5-6-7-8-9-10-9-11-12-13

Camino 5: 1-2-3-4-5-6-7-8-9-10-11-8-9-10-11-12-13

Resultados Obtenidos:

Se analizaron 3 métodos, correspondientes a las funcionalidades de mayor importancia en el módulo. En una primera iteración resultaron satisfactorios los métodos **eventos ()** y **jsondata ()**, representando el 66.6% del total. El método **save ()** arrojó error en una de sus estructura *if*, representando el 33,4% del total. En una segunda iteración el código fue corregido, representando el 100% de las pruebas satisfactorias.

3.2 Pruebas de Aceptación

Las pruebas de aceptación son pruebas de caja negra definidas por el cliente para cada HU, y tienen como objetivo asegurar que las funcionalidades del sistema cumplen con lo que se espera de ellas. En efecto, las pruebas de aceptación corresponden a una especie de documento de requerimientos en XP, ya que marcan el camino a seguir en cada iteración, indicándole al equipo de desarrollo hacia donde tiene que ir y en qué puntos o funcionalidades debe poner el mayor esfuerzo y atención (34).

Como resultado de las pruebas de aceptación se obtendrán artefactos descritos en tablas, estas contarán con los siguientes campos:

- **Código:** servirá como identificador de la prueba realizada, a su vez será sugerente al nombre de la prueba a la que hace referencia.
- **HU:** tendrá el nombre de la HU a la que hace referencia la prueba a realizar.
- **Nombre:** nombre que se le da a la prueba a realizar.
- **Descripción:** se describe la funcionalidad que se desea probar.
- **Condiciones de Ejecución:** mostrará las condiciones que deben cumplirse para poder llevar a cabo el caso de prueba, estas condiciones deben ser satisfechas antes de la ejecución del caso de prueba para que se puedan obtener los resultados esperados.
- **Entradas/Pasos de Ejecución:** se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las entradas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado.
- **Resultado esperado:** se hará una breve descripción del resultado que se espera obtener con la prueba realizada.
- **Evaluación de la prueba:** acorde al resultado de la prueba realizada se emitirá una evaluación sobre la misma. Esta evaluación tendrá uno de los tres resultados que a continuación se describen:
 - **Bien:** cuando el resultado de la prueba es exactamente el esperado por el usuario.
 - **Parcialmente bien:** cuando el resultado no es completamente el esperado por el cliente o usuario de la aplicación y muestra resultados erróneos o fuera de contexto.
 - **Mal:** cuando el resultado de la prueba realizada genera un error de codificación en la aplicación o muestra como resultado elementos no deseados o fuera de contexto, trayendo como consecuencia que la funcionalidad requerida por el cliente no tenga resultado, lo que invalida también la HU.

A continuación se muestran unas de las pruebas de aceptación más relevantes y las restantes se pueden visualizar en el [Anexo 3](#).

Caso de Prueba de Aceptación

Código: HU1_P1	Historia de Usuario: 1
Nombre: Autenticar usuario	
Descripción: Prueba para la funcionalidad de autenticar usuario	
Condiciones de Ejecución: Se debe acceder a la interfaz principal para acceder a los datos del sistema.	
Entradas/Pasos de Ejecución: Se llenan los campos según su Email y su correspondiente contraseña, luego se opta por la opción ENTRAR. El sistema valida que la entrada de su contraseña se corresponda al Email ingresado; que estas sean correctas.	
Resultado Esperado: Se autentica de forma correcta el usuario.	
Evaluación de la Prueba: Bien.	

Tabla 9. Prueba de Aceptación#1 Autenticar Usuario.

Caso de Prueba de Aceptación	
Código: HU6_P6	Historia de Usuario: 6
Nombre: Gestionar evento.	
Descripción: Prueba para la funcionalidad gestionar los eventos que se producen en el sistema.	
Condiciones de Ejecución: Se debe acceder a la interfaz que muestra el listado de los eventos y seleccionar uno de ellos.	
Entradas/Pasos de Ejecución: Del listado de los eventos existentes, se puede seleccionar la opción adicionar, se validan las entradas y se retorna a la interfaz inicial. Se puede además seleccionar un evento para su posterior modificación, así como eliminarlo. También se pueden realizar búsquedas de los eventos existentes en el sistema, según parámetros seleccionados. El sistema tras toda acción, valida que las entradas sean correctas.	
Resultado Esperado: Se gestiona correctamente el evento.	
Evaluación de la Prueba: Bien.	

Tabla 10. Prueba de Aceptación#2 Gestionar Evento.

Caso de Prueba de Aceptación

Código: HU2_P2	Historia de Usuario: 2
Nombre: Gestionar usuario.	
Descripción: Prueba para la funcionalidad de gestionar al usuario que pertenecerá a un evento, ya sea crearlos, buscarlos, modificar sus parámetros y eliminarlos.	
Condiciones de Ejecución: Se debe acceder a la interfaz que muestra el listado de los usuarios pertenecientes al evento.	
Entradas/Pasos de Ejecución: Del listado de los usuarios existentes, se puede seleccionar la opción adicionar, se validan las entradas y se retorna a la interfaz inicial. También se pueden realizar búsquedas de usuarios existentes en el sistema, según parámetros seleccionados. Además esta lo opción de modificarlos, así como eliminarlos. El sistema tras toda acción del usuario valida que las entradas sean correctas.	
Resultado Esperado: Se gestiona correctamente el usuario.	
Evaluación de la Prueba: Bien.	

Tabla 11. Prueba de Aceptación#3 Gestionar Usuario.

Resultados Obtenidos:

Se realizaron 3 iteraciones para 14 pruebas funcionales internas, arrojando como resultado 17 no conformidades. En la primera iteración fueron clasificadas de bien 2 de ellas, representando un 14,29% del total y 12 mal, para un 85,71% del total. En una segunda iteración fueron clasificadas de bien 9 de ellas, representando un 64,28% del total y 5 mal, representando un 35,72% del total. En una tercera y última iteración resultaron todas satisfactorias, constituyendo un 100% de pruebas funcionales exitosas.

3.2 Conclusiones

Las pruebas unitarias realizadas a las funcionalidades claves de la aplicación, permitieron corregir errores del código de forma independiente. La utilización de las pruebas de aceptación permitió comprobar el cumplimiento de todos los requisitos funcionales establecidos. Realizada la validación de la solución propuesta mediante los distintos casos de pruebas diseñados, se obtuvo que el módulo se encuentra listo y con la calidad requerida satisfaciendo todas las exigencias del cliente.

CONCLUSIONES

- El análisis de los sistemas existentes en el mundo que implementan procesos de la Gestión de Servicios según ITIL, arrojó que ninguno de los encontrados responde a la Gestión de Eventos.
- El análisis de la metodología XP propició a los desarrolladores implementar el módulo garantizando la generación de artefactos y documentación necesaria.
- El empleo del patrón de arquitectura MVC permitió que la interfaz usuario y la lógica del negocio se implementaran de forma independiente.
- Los resultados obtenidos durante la validación conformada demostraron el correcto funcionamiento del módulo de Gestión de Eventos.
- El módulo desarrollado brinda a los especialistas del Centro de Soporte UCI minimizar el tiempo de respuesta del servicio al cliente.

RECOMENDACIONES

Luego de cumplir con cada una de las tareas de la investigación y partiendo de la experiencia acumulada en el desarrollo de la misma se proponen las siguientes recomendaciones:

- Continuar ampliando el módulo Gestión de Eventos, según necesidades futuras del Centro de Soporte UCI.
- Creación de un Manual de Usuario del módulo Gestión de Eventos para mejorar la navegación del usuario en el sistema.
- La implementación de la autenticación LDAP en el módulo Gestión de Eventos, para la posterior integración al sistema Gestión de Servicios del Centro de Soporte UCI.

REFERENCIAS BIBLIOGRÁFICAS

1. **Alter.** *¿Qué es la Tecnología de la Información?* 1999.
2. **Bologna, Walsh.** *Tecnología de la Información.* 1997.
3. Gestión de Servicios. *Gestión de Servicios.* [En línea] 2013. [Citado el: 12 de Octubre de 2013.] [\]http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/fundamentos_de_la_gestion_TI/vision_general_gestion_servicios_TI/vision_general_gestion_servicios_TI.php](http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/fundamentos_de_la_gestion_TI/vision_general_gestion_servicios_TI/vision_general_gestion_servicios_TI.php).
4. Soporte Informáticos. *Soporte Informáticos.* [En línea] [Citado el: 15 de Octubre de 2013.] <http://www.mikogo.es/guia/soporte-informatico/>.
5. Gestión de eventos in Curso ITIL v3. [En línea] [Citado el: 17 de Octubre de 2013.]
6. DefinicionesABC. [En línea] 2011. [Citado el: 17 de Octubre de 2013.] <http://www.definicionabc.com/economia/servicio.php..>
7. *PMI.Guía de los fundamentos de la Dirección de Proyectos.* Pennsylvania : Project Management Institute, 2004.
8. **Thompson, Ivan.** Definiciones de Servicios. [En línea] Agosto de 2006. [Citado el: 10 de Enero de 2013.] <http://www.promonegocios.net/mercadotecnia-servicios/definicion-servicios.html..>
9. Glosario y abreviaturas de ITIL Español (Latinoamericano). [En línea] [Citado el: 28 de Octubre de 2013.] <http://www.official-site.com>.
10. *Aplicación de la norma COBIT en el monitoreo de transferencias.* **Arias, Aymara Hernández y Graterol, Carolina Benavides.** s.l. : Publicaciones en Ciencias y Tecnologías, 2011, Vol. 5. 1856-8890.
11. **Cohello, Helkyn.** Como integrar y adoptar los estándares para un buen Gobierno de TI. [En línea] 8 de Diciembre de 2008. [Citado el: 10 de Diciembre de 2013.] [http://helkyncoello.wordpress.com/2008/12/08/itil-cobit-cmmi-pmbok-como-integrar-y-adoptar-los-estandares-para-un-buen-gobierno-de-ti/..](http://helkyncoello.wordpress.com/2008/12/08/itil-cobit-cmmi-pmbok-como-integrar-y-adoptar-los-estandares-para-un-buen-gobierno-de-ti/)

12. **Sanchez, Joaquín Trejo y Corona, Eduardo Reyes.** Calidad en TI/ISO 20000. [En línea] Universidad Autónoma de la ciudad de México, 2014. [Citado el: 18 de Enero de 2014.] http://www.calidadti.cetecna.com/?page_id=37.
13. ManageEngineServiceDeskPlus. [En línea] [Citado el: 12 de Marzo de 2014.] <http://www.manageengine.com/products/service-desk/spanish/>.
14. Trellis Desk. [En línea] [Citado el: 16 de Octubre de 2013.] <http://www.slideboom.com/presentations/318246/Curso-ITIL-ISO20000>.
15. Kayako. [En línea] [Citado el: 18 de Octubre de 2013.] <http://forums.kayako.com/threads/itil-module-for-kayako.36479/>.
16. Cerberus. [En línea] [Citado el: 18 de Octubre de 2013.] <http://www.linkedin.com/groups/Does-anyone-use-JIRA-ITIL-51930.S.45121829>.
17. CA. [En línea] [Citado el: 20 de Octubre de 2013.] <http://www.ca.com/us/itil-process-maps.aspx>.
18. **Calderón, Amaro y Sarah, Damaris.** *Metodologías Ágiles*. Trujillo, Perú : s.n., 2007.
19. Microsoft ASP.NET. [En línea] [Citado el: 30 de Enero de 2014.] <http://www.asp.net/mvc>.
20. VisualParadimg. [En línea] [Citado el: 26 de Enero de 2014.] [http://www.visual-paradigm.com/..](http://www.visual-paradigm.com/)
21. Grails. [En línea] 2012. [Citado el: 15 de Enero de 2014.] [http://grails.bandcamp.com/..](http://grails.bandcamp.com/)
22. Groovy A Dynamic Language. [En línea] [Citado el: 30 de Enero de 2014.] [http://groovy.codehaus.org/..](http://groovy.codehaus.org/)
23. **Darcy, Hamlet.** JetBrains Intellig. [En línea] [Citado el: 30 de Enero de 2014.] <http://www.jetbrains.org/display/IJOS/Home;jsessionid=689081AE72E3750157BF04901E9916A1>.
24. PostgreSQL. [En línea] [Citado el: 30 de Enero de 2014.] <http://www.postgresql.org/>.
25. **Sommerville, Ian.** *Ingeniería de Software*. Madrid : Pearson Educación S.A, 2005. 84-7829-074-5.
26. **Garzas, Javier.** JavierGarzas.com. [En línea] 22 de Diciembre de 2011. [Citado el: 22 de Febrero de 2014.] <http://www.javiergarzas.com/2011/12/historia-de-usuario-diferente-de-requisito.html>.
27. **Echeberry, Luis Miguel Tobon y Delgado, Carmona Luz Elena.** *Caso práctico de la metodología XP al desarrollo de Software*. s.l. : Pereira, 2007.

28. **Malfará, Dayvis.** *Testing en eXtreme Programing.* 2006. 3860676-3.
29. Patrón de Diseño. [En línea] [Citado el: 14 de Marzo de 2014.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
30. **Cortes, Gloria y Casallas, Rubby.** *Introducción a los patrones de Software.* Colombia : Universidad de los Andes, 2004.
31. PatrónMVC. [En línea] [Citado el: 12 de Marzo de 2014.] <http://www.tab.inf.uc3m.es/~a0080802/RAI/mvc.html>.
32. **Miranda, Marco.** *Documento de Estándares de Programación.* 2010.
33. **Pressman, Roger.** *Ingeniería de Software:Un enfoque práctico.* México : MCGRAW- HILL, 2005. 9789701054734.
34. **Gutierrez, J.J, y otros.** *Pruebas del sistema en programación.* Sevilla : s.n.
35. **Marca Huallpara Hugo Michael, Quisbert Limachi Nancy Susana.** Diagrama de Despliegue.doc.[En línea] [Citado el: 16 de junio de 2014]virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc.

BIBLIOGRAFÍA

Alter. *¿Qué es la Tecnología de la Información?* 1999.

Bologna, Walsh. *Tecnología de la Información.* 1997.

Gestión de Servicios. *Gestión de Servicios.* [En línea] 2013. [Citado el: 12 de Octubre de 2013.] http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/fundamentos_de_la_gestion_TI/vision_general_gestion_servicios_TI/vision_general_gestion_servicios_TI.php.

Soporte Informáticos. *Soporte Informáticos.* [En línea] [Citado el: 15 de Octubre de 2013.] <http://www.mikogo.es/guia/soporte-informatico/>.

Gestión de eventos in Curso ITIL v3. [En línea] [Citado el: 17 de Octubre de 2013.]

DefinicionesABC. [En línea] 2011. [Citado el: 17 de Octubre de 2013.] <http://www.definicionabc.com/economia/servicio.php..>

PMI.Guía de los fundamentos de la Dirección de Proyectos. Pennsylvania : Project Management Institute, 2004.

Thompson, Ivan. Definiciones de Servicios. [En línea] Agosto de 2006. [Citado el: 10 de Enero de 2013.] <http://www.promonegocios.net/mercadotecnia-servicios/definicion-servicios.html..>

Glosario y abreviaturas de ITIL Español (Latinoamericano). [En línea] [Citado el: 28 de Octubre de 2013.] <http://www.official-site.com>.

Aplicación de la norma COBIT en el monitoreo de transferencias. **Arias, Aymara Hernández y Graterol, Carolina Benavides.** s.l. : Publicaciones en Ciencias y Tecnologías, 2011, Vol. 5. 1856-8890.

Cohello, Helkyn. Como integrar y adoptar los estándares para un buen Gobierno de TI. [En línea] 8 de Diciembre de 2008. [Citado el: 10 de Diciembre de 2013.] [http://helkyncoello.wordpress.com/2008/12/08/itil-cobit-cmmi-pmbok-como-integrar-y-adoptar-los-estandares-para-un-buen-gobierno-de-ti/..](http://helkyncoello.wordpress.com/2008/12/08/itil-cobit-cmmi-pmbok-como-integrar-y-adoptar-los-estandares-para-un-buen-gobierno-de-ti/)

Sanchez, Joaquín Trejo y Corona, Eduardo Reyes. Calidad en TI/ISO 20000. [En línea] Universidad Autónoma de la ciudad de México, 2014. [Citado el: 18 de Enero de 2014.] http://www.calidadti.cetecna.com/?page_id=37.

ManageEngineServiceDeskPlus. [En línea] [Citado el: 12 de Marzo de 2014.] <http://www.manageengine.com/products/service-desk/spanish/>.

Trellis Desk. [En línea] [Citado el: 16 de Octubre de 2013.] <http://www.slideboom.com/presentations/318246/Curso-ITIL-ISO20000>.

Kayako. [En línea] [Citado el: 18 de Octubre de 2013.] <http://forums.kayako.com/threads/itil-module-for-kayako.36479/>.

Cerberus. [En línea] [Citado el: 18 de Octubre de 2013.] <http://www.linkedin.com/groups/Does-anyone-use-JIRA-ITIL-51930.S.45121829>.

CA. [En línea] [Citado el: 20 de Octubre de 2013.] <http://www.ca.com/us/itil-process-maps.aspx>.

Calderón, Amaro y Sarah, Damaris. *Metodologías Ágiles*. Trujillo, Perú : s.n., 2007.

Microsoft ASP.NET. [En línea] [Citado el: 30 de Enero de 2014.] <http://www.asp.net/mvc>.

VisualParadimg. [En línea] [Citado el: 26 de Enero de 2014.] [http://www.visual-paradigm.com/..](http://www.visual-paradigm.com/)

Grails. [En línea] 2012. [Citado el: 15 de Enero de 2014.] [http://grails.bandcamp.com/..](http://grails.bandcamp.com/)

Groovy A Dynamic Language. [En línea] [Citado el: 30 de Enero de 2014.] [http://groovy.codehaus.org/..](http://groovy.codehaus.org/)

Darcy, Hamlet. JetBrains Intellig. [En línea] [Citado el: 30 de Enero de 2014.] <http://www.jetbrains.org/display/IJOS/Home;jsessionid=689081AE72E3750157BF04901E9916A1>.

PostgreSQL. [En línea] [Citado el: 30 de Enero de 2014.] <http://www.postgresql.org/>.

Sommerville, Ian. *Ingeniería de Software*. Madrid : Pearson Educación S.A, 2005. 84-7829-074-5.

Garzas, Javier. JavierGarzas.com. [En línea] 22 de Diciembre de 2011. [Citado el: 22 de Febrero de 2014.] <http://www.javiergarzas.com/2011/12/historia-de-usuario-diferente-de-requisito.html>.

Echeberry, Luis Miguel Tobon y Delgado, Carmona Luz Elena. *Caso práctico de la metodología XP al desarrollo de Software*. s.l. : Pereira, 2007.

Malfará, Dayvis. *Testing en eXtreme Programing*. 2006. 3860676-3.

Patrón de Diseño. [En línea] [Citado el: 14 de Marzo de 2014.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.

Cortes, Gloria y Casallas, Rubby. *Introducción a los patrones de Software.* Colombia : Universidad de los Andes, 2004.

PatrónMVC. [En línea] [Citado el: 12 de Marzo de 2014.]
<http://www.tab.inf.uc3m.es/~a0080802/RAI/mvc.html>.

Miranda, Marco. *Documento de Estándares de Programación.* 2010.

Pressman, Roger. *Ingeniería de Software:Un enfoque práctico.* México : MCGRAW- HILL, 2005.
9789701054734.

Gutierrez, J.J, y otros. *Pruebas del sistema en programación.* Sevilla : s.n.

ANEXOS

Anexo 1. Historias de Usuarios.

Historia de Usuario	
Número:3	Nombre de la Historia de Usuario: Gestionar grupo de usuarios
Cantidad de modificaciones a la Historia de Usuario: Ninguna	
Usuario: Administrador	Iteración asignada: 1
Prioridad en negocio: Muy Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Descripción: <ol style="list-style-type: none"> 1. El sistema muestra la interfaz principal. 2. El administrador introduce sus datos con su contraseña. 3. El sistema muestra una interfaz según sus permisos administrativos. 4. El administrador pulsa en la opción "Grupos de Usuarios". 5. El sistema muestra el listado de los grupos existentes en el sistema. 6. El administrador modifica el nombre del grupo según convenga. 7. El sistema muestra un cuadro de diálogo, informándole al administrador que el grupo ha sido modificado correctamente. 	
Observaciones: Si los datos de modificación del grupo de usuarios no son válidos la aplicación mostrará mensajes con los errores correspondientes.	
Prototipo de interfaces:	

Criterios de búsqueda

Nombre:

[+ Adicionar](#)

Listado de grupos

NOMBRE

Administrador



Jefe de Grupo



Supervisor



Usuario



Adicionar grupo de usuarios

DATOS DEL NUEVO GRUPO

Nombre

Modificar grupo de usuarios

DATOS DEL GRUPO A MODIFICAR

Nombre

Tabla 12.HU#3 Gestionar Grupo de Usuarios.

Historia de Usuario	
Número: 4	Nombre de la Historia de Usuario: Gestionar categoría del evento.
Cantidad de modificaciones a la Historia de Usuario: Ninguna	
Usuario: Supervisor	Iteración asignada: 1
Prioridad en negocio: Muy Alta	Puntos estimados: 1 y½ semana
Riesgo en desarrollo: Alto	Puntos reales: 1 y ½ semana
<p>Descripción:</p> <ol style="list-style-type: none"> 1. El sistema muestra la interfaz correspondiente a la gestión de categoría. 2. El administrador pulsa en la opción “Adicionar”. 3. El sistema muestra una interfaz que le permitirá al administrador adicionar una categoría de los eventos. 4. El administrador adiciona la categoría llenando los campos correspondientes. 5. Si el administrador desea modificar una categoría, se selecciona una de las que se encuentran en el listado del sistema y se pulsa la opción . 	

Observaciones: -

Prototipo de interfaces:

Criterios de búsqueda

Nombre:

 Adicionar

Listado de categorías

TIPO

Alerta		
Excepción		
Informativo		

Adicionar categoría

DATOS DE NUEVA CATEGORÍA

Tipo

Modificar categoría

DATOS DE CATEGORÍA A MODIFICAR

Tipo

Tabla 13.HU#4 Gestionar Categoría

Historia de Usuario	
Número: 5	Nombre de la Historia de Usuario: Gestionar sub_categoría del evento.
Cantidad de modificaciones a la Historia de Usuario: Ninguna	
Usuario: Supervisor	Iteración asignada: 1
Prioridad en negocio: Muy Alta	Puntos estimados: 1 y ½ semana
Riesgo en desarrollo: Alto	Puntos reales: 1 y½ semana
<p>Descripción:</p> <ol style="list-style-type: none"> 1. El sistema muestra la interfaz correspondiente a la gestión de sub_ categoría. 2. El administrador pulsa en la opción “Adicionar”. 3. El sistema muestra una interfaz que le permitirá al administrador adicionar una sub_categoría de los eventos. 4. El administrador adiciona la sub_categoría llenando los campos correspondientes. 5. Si el administrador desea modificar una sub_categoría, se selecciona una de las que se encuentran en el listado del sistema y se pulsa la opción . 	

Observaciones: El administrador solo podrá adicionar eventos a la sub_categoría que esté insertando en el sistema.

Prototipo de interfaces:

Criterios de búsqueda

Nombre:

[+ Adicionar](#)

Listado de subcategorías

TIPO	
Alerta de Mantenimiento	 

Adicionar subcategoría

DATOS DE NUEVA SUBCATEGORÍA

Nombre

Tabla 14.HU#5 Gestionar sub_categoria

Anexo 2. Tarjetas CRC.

Tarjeta CRC	
Clase: Categoría	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> • Es la encargada de almacenar los atributos que requiere una categoría de elemento de la configuración: <ul style="list-style-type: none"> ○ String tipo_categoria 	

Tabla 15.TarjetaCRC#3

Tarjeta CRC
Clase: CategoríaController

Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> • Es la encargada de: <ul style="list-style-type: none"> ○ index() ○ editPage() ○ addPage() • Además de las funcionalidades que genera de forma automática el framework: <ul style="list-style-type: none"> ○ list(Integermax) ○ create() ○ save() ○ show(Long id) ○ edit(Long id) ○ update(Long id, Long version) ○ delete(Long id) 	<ul style="list-style-type: none"> ○ Categoría

Tabla 16. Tarjeta CRC#4

Tarjeta CRC	
Clase: Usuario	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> • Es la encargada de almacenar los atributos que requiere un cliente de elemento de la configuración: <ul style="list-style-type: none"> ○ String nombre ○ String correo ○ String contraseña ○ String rol 	

○ String ci	
-------------	--

Tabla 17. Tarjeta CRC#5

Tarjeta CRC	
Clase: Sub_categoría	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> • Es la encargada de almacenar los atributos que requiere una sub_categoría de elemento de la configuración: <ul style="list-style-type: none"> ○ String tipo_sub_categoría 	

Tabla 18. Tarjeta CRC#7

Tarjeta CRC	
Clase: Sub_CategoríaController	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> • Es la encargada de: <ul style="list-style-type: none"> ○ index() • Además de las funcionalidades que genera de forma automática el framework: <ul style="list-style-type: none"> ○ list(Integer max) ○ create() ○ save() ○ show(Long id) ○ edit(Long id) ○ update(Long id, Long version) 	<ul style="list-style-type: none"> ○ Sub_Categoría

○ delete(Long id)	
-------------------	--

Tabla 19.Tarjeta CRC#8

Tarjeta CRC	
Clase: GrupoUsuario	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> • Es la encargada de almacenar los atributos que requiere un grupo de usuarios de elemento de la configuración: <ul style="list-style-type: none"> ○ String rol 	

Tabla 20.TarjetaCRC#9

Tarjeta CRC	
Clase: Impacto	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> • Es la encargada de almacenar los atributos que requiere un impacto de elementos de la configuración: <ul style="list-style-type: none"> ○ String nombre 	

Tabla 21.Tarjeta CRC#10

Tarjeta CRC

Clase: Notificación	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> • Es la encargada de almacenar los atributos que requiere una notificación de elemento de la configuración: <ul style="list-style-type: none"> ○ String sender ○ String catcher ○ String subject ○ String message ○ Boolean state 	

Tabla 22. Tarjeta CRC#11

Tarjeta CRC	
Clase: Actividad	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> • Es la encargada de almacenar los atributos que requiere una urgencia de elemento de la configuración: <ul style="list-style-type: none"> ○ String nombre 	

Tabla 23. Tarjeta CRC#12

Tarjeta CRC	
Clase: Área	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none"> • Es la encargada de almacenar los atributos que requiere una urgencia de elemento de la configuración: <ul style="list-style-type: none"> ○ String nombre 	

Tabla 24. Tarjeta CRC#13

Anexo 3. Pruebas de Aceptación.

Caso de Prueba de Aceptación	
Código: HU3_P3	Historia de Usuario: 3
Nombre: Gestionar grupo de usuarios.	
Descripción: Prueba para la funcionalidad de gestionar grupo de usuarios que pertenecerá a un evento, ya sea crearlos, modificar sus parámetros y eliminarlos.	
Condiciones de Ejecución: Se debe acceder a la interfaz que muestra el listado de los grupos de usuarios pertenecientes al evento.	
Entradas/Pasos de Ejecución: Del listado de los eventos existentes, se selecciona un evento, luego el sistema muestra el listado de los usuarios mostrándose al grupo al que estos pertenecen del evento correspondiente. Del listado de los grupos de usuarios existentes, se puede seleccionar la opción adicionar, se validan las entradas y se retorna a la interfaz inicial. Se puede además seleccionar un grupo para su posterior modificación, así como eliminarlo. El sistema tras toda acción valida que las entradas sean correctas.	
Resultado Esperado: Se gestiona correctamente el grupo de usuarios.	
Evaluación de la Prueba: Bien.	

Tabla 25. Prueba de Aceptación #3

Caso de Prueba de Aceptación	
Código: HU4_P4	Historia de Usuario: 4
Nombre: Gestionar categoría	
Descripción: Prueba para la funcionalidad de gestionar una categoría de un evento, ya sea crearlas, modificar sus parámetros y eliminarlas.	
Condiciones de Ejecución: Se debe acceder a la interfaz que muestra el listado de las categorías pertenecientes al evento.	
Entradas/Pasos de Ejecución: Del listado de los eventos existentes, se selecciona	

un evento, luego el sistema muestra la categoría a la q pertenece el mismo. Si se desea modificar dicha categoría se opta la opción modificar y se muestran las demás categorías existentes para los eventos y se realiza la modificación o también su posterior eliminación. Del listado de las categorías existentes, se puede seleccionar la opción adicionar, se validan las entradas y se retorna a la interfaz inicial. El sistema tras toda acción del usuario valida que las entradas sean correctas.

Resultado Esperado: Se gestionan correctamente las categorías.

Evaluación de la Prueba: Bien.

Tabla 26. Prueba de Aceptación #4

Caso de Prueba de Aceptación	
Código: HU5_P5	Historia de Usuario: 5
Nombre: Gestionar sub_categoría	
Descripción: Prueba para la funcionalidad de gestionar una sub_categoría de un evento, ya sea crearlas, modificar sus parámetros y eliminarlas.	
Condiciones de Ejecución: Se debe acceder a la interfaz que muestra el listado de las sub_categoría pertenecientes a la categoría del evento.	
Entradas/Pasos de Ejecución: Del listado de los eventos existentes, se selecciona un evento, luego el sistema muestra la categoría a la q pertenece el mismo y además la sub_categoría que esta tiene dentro de la misma. Si se desea modificar dicha sub_categoría se opta la opción modificar y se muestran las demás sub_categorías existentes para los eventos y se realiza la modificación o también su posterior eliminación. Del listado de las sub_categorías existentes, se puede seleccionar la opción adicionar, se validan las entradas y se retorna a la interfaz inicial. El sistema tras toda acción del usuario valida que las entradas sean correctas.	
Resultado Esperado: Se gestionan correctamente las sub_categorías.	
Evaluación de la Prueba: Bien.	

Tabla 27. Prueba de Aceptación #5

Caso de Prueba de Aceptación

Código: HU7_P7	Historia de Usuario: 7
Nombre: Visualizar impacto del evento.	
Descripción: Prueba para la funcionalidad de visualizar el impacto de un evento determinado.	
Condiciones de Ejecución: Se debe acceder a la interfaz que muestra el listado de los eventos y seleccionar uno.	
Entradas/Pasos de Ejecución: Del listado de los eventos existentes tras seleccionar uno de ellos se opta por la opción de Detalles del Evento y se mostrará el impacto, además de otros detalles del evento. El sistema tras toda acción, valida que las entradas sean correctas	
Resultado Esperado: Se visualiza correctamente el impacto y los procesos afectados del evento.	
Evaluación de la Prueba: Bien.	

Tabla 28. Prueba de Aceptación #7

Caso de Prueba de Aceptación	
Código: HU8_P8	Historia de Usuario: 8
Nombre: Visualizar procesos afectados.	
Descripción: Prueba para la funcionalidad de mostrar procesos afectados.	
Condiciones de Ejecución: Se debe acceder a la interfaz que muestra el listado de los eventos.	
Entradas/Pasos de Ejecución: Luego de acceder al listado de los eventos existentes, luego se opta por Matriz de Eventos, y se realiza la acción a ejecutar, visualizando los procesos que afectan los eventos seleccionados.	
Resultado Esperado: Se muestra correctamente los procesos afectados.	
Evaluación de la Prueba: Bien.	

Tabla 29. Prueba de Aceptación #8

Caso de Prueba de Aceptación	
Código: HU9_P9	Historia de Usuario: 9

Nombre: Mostrar Matriz de eventos.
Descripción: Prueba para la funcionalidad de mostrar matriz de eventos.
Condiciones de Ejecución: Se debe acceder a la interfaz que muestra el listado de los eventos.
Entradas/Pasos de Ejecución: Luego de acceder al listado de los eventos existentes, luego se opta por Matriz Eventos y se realiza la acción a ejecutar.
Resultado Esperado: Se muestra correctamente la matriz de eventos.
Evaluación de la Prueba: Bien.

Tabla 30. Prueba de Aceptación#9

Caso de Prueba de Aceptación	
Código: HU10_P10	Historia de Usuario: 10
Nombre: Permitir realizar notificaciones de eventos a personas.	
Descripción: Prueba para la funcionalidad permitir realizar notificaciones de eventos a personas.	
Condiciones de Ejecución: Se debe acceder a la interfaz que muestra el listado de los eventos.	
Entradas/Pasos de Ejecución: En el listado de eventos existentes, se selecciona la opción de Realizar Notificación. El sistema muestra la cantidad de notificaciones pertenecientes a dicho evento y el usuario selecciona la opción de enviar o no la notificación. Además el usuario podrá enviar tantas notificaciones como sea necesario.	
Resultado Esperado: Se realizan correctamente las notificaciones.	
Evaluación de la Prueba: Bien.	

Tabla 26. Prueba de Aceptación #10

Caso de Prueba de Aceptación	
Código: HU11_P11	Historia de Usuario: 11
Nombre: Permite exportar el listado de eventos formato pdf.	
Descripción: Prueba para la funcionalidad de exportar en formato pdf el listado de eventos.	

Condiciones de Ejecución: Se debe acceder a la interfaz que muestra el listado de los eventos.
Entradas/Pasos de Ejecución: Al mostrar el listado de los eventos, se selecciona la opción Exportar Listado. El sistema debe muestra toda la información que describe al evento en formato pdf.
Resultado Esperado: Se exporta correctamente el listado de los eventos.
Evaluación de la Prueba: Bien.

Tabla 31. Prueba de Aceptación #11

Caso de Prueba de Aceptación	
Código: HU12_P12	Historia de Usuario: 12
Nombre: Listar procesos que pudieran ser afectados por los eventos.	
Descripción: Prueba para la funcionalidad de listar procesos que pudieran estar afectados por los eventos.	
Condiciones de Ejecución: Se debe acceder a la interfaz que muestra el listado de los eventos y se selecciona uno de ellos.	
Entradas/Pasos de Ejecución: En el listado de los eventos, se selecciona uno de ellos por medio de Detalles del Evento, se mostrará los procesos que han sido afectados por la ejecución de dicho evento, además de otros detalles del evento.	
Resultado Esperado: Se muestra correctamente el listado de los procesos afectados por el evento seleccionado.	
Evaluación de la Prueba: Bien.	

Tabla 32. Prueba de Aceptación #12

Caso de Prueba de Aceptación	
Código: HU13_P13	Historia de Usuario: 13
Nombre: Realizar búsquedas por parámetros.	
Descripción: Prueba para la funcionalidad de realizar búsquedas por parámetros.	
Condiciones de Ejecución: Se debe acceder a la interfaz que muestra el listado de los eventos.	
Entradas/Pasos de Ejecución: Al mostrar el listado de los eventos, se desea realizar	

una búsqueda por parámetros específicos, luego de filtrado dichos parámetros se muestran los evento(s) del sistema q cumple(n) con los mismos.

Resultado Esperado: Se realiza las búsquedas por parámetros correctamente.

Evaluación de la Prueba: Bien.

Tabla 33. Prueba de Aceptación #13

Caso de Prueba de Aceptación	
Código: HU14_P14	Historia de Usuario: 14
Nombre: Permite visualizar gráficas estadísticas de los eventos.	
Descripción: Prueba para la funcionalidad de visualizar gráficas estadísticas de los eventos.	
Condiciones de Ejecución: Se debe acceder a la interfaz que muestra el listado de los eventos.	
Entradas/Pasos de Ejecución: Al mostrar el listado de los eventos, se selecciona la opción Gráficas Estadísticas. El sistema muestra la gráfica estadísticas de los eventos existentes en el sistema.	
Resultado Esperado: Se visualiza correctamente la gráfica estadística de los eventos.	
Evaluación de la Prueba: Bien.	

Tabla 34. Prueba de Aceptación #15

Anexo4. Entrevista.

Entrevista

Realizada dos especialistas.

Objetivo: comprobar cómo se desarrollan la seguridad y la confidencialidad en los Departamentos de Seguridad Social y Prevención y Atención y Asistencia Social.

Compañero o compañera.

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

Datos generales.

Especialista #1:

✓ Nombre y Apellidos: Yulio Seriocha García Gallardo.

✓ Años de experiencia: 6 años.

Especialista #2:

✓ Nombre y Apellidos: Neybis Lago Clara.

✓ Años de experiencia: 2 años.

Aspecto a encuestar

1. La información que se maneja en los departamentos ¿es de carácter sensible? ¿Por qué?

Respuesta: Si, porque se pone en riesgo la información que se maneja en el centro de los diferentes software de la universidad a los que se le da soporte.

2. Cuando se desea acceder a la información que se maneja en el departamento cualquier persona puede realizar dicho procesos. Argumente.

Respuesta: No, porque las personas tienen disponibilidad de los datos según el rol que ocupe en el centro.

3. ¿Qué cantidad de personas del departamento tienen acceso a la información que se maneja?

Respuesta: A la gestión de eventos tienen acceso todos, al menos a verlos y crearlos, (alrededor de 15 especialistas). Y a modificarlos y eliminarlos 3 especialistas.

4. ¿Cuál o cuáles son las vías que se utilizan para realizar las entregas de información desde las diferentes entidades que se vinculan al departamento?

Respuesta: En dependencia del proceso, si es Gestión de Incidencia y Problema, a través de la plataforma. Los Acuerdos de nivel de servicio y la Gestión de Cambio, en formato duro o vía correo.

5. ¿Qué beneficios encuentra usted a la digitalización de la información para tratarla de manera más centralizada?

Respuesta: Esta siempre disponible en la red, es más segura y existe menos probabilidad de pérdida de la información.

Anexo5. Aval del Cliente.

**CENTRO DE
SOPORTE**

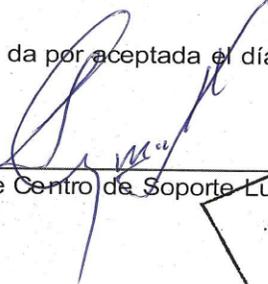


Por este medio avalo que la tesis de grado "Módulo de Gestión de Eventos para el Centro de Soporte de la Universidad de las Ciencias Informáticas" de las tesis Amanda Gómez Marrero y Tatiana Miranda Terrero es de gran importancia para el Centro de Soporte y la aplicación generada formará parte de la Plataforma de Gestión de Servicio. El sistema cuenta con los requerimientos especificados por el cliente, así como un buen diseño. Además cumple con lo establecido por la metodología de gestión de servicio.

Sobre el Trabajo de Diploma realizado considero que favorece la calidad de la gestión de servicio, sirviendo de soporte y ayuda para los desarrolladores del proyecto productivo que dará continuación al trabajo presentado, sentando las bases para la futura construcción de la plataforma concebida.

Los diplomantes han puesto en evidencia los conocimientos adquiridos siendo capaces de ponerlos en práctica para solucionar la problemática planteada. Han mantenido una aptitud muy responsable y comprometida con la investigación, logrando realizar un excelente trabajo individual y en equipo.

Y para que así conste se da por aceptada el día 28 del mes 05 del año 2014 por:

Director de Centro de Soporte Luis J.  Juan Hernández

GLOSARIO DE TÉRMINOS

Ciclo de Vida: Son las diferentes etapas en la vida de un servicio de las TI, CI, incidente, problema y cambio. El ciclo de vida define las categorías por estado y las transiciones de estado permitidas

Concurrencia: Es una medida del número de usuarios que están participando al mismo tiempo en una misma sesión.

Gestión de Eventos: Es el proceso responsable de gestionar los eventos durante todo su ciclo de vida. La gestión de eventos es una de las principales actividades de las operaciones de las TI.

Incidente: Es una interrupción no planificada de un servicio de las TI o la reducción en la calidad de un servicio de las TI. La falla de un elemento de configuración que no ha afectado aún el servicio es también un incidente - por ejemplo, la falla en un disco de un conjunto de discos espejos.

Infraestructura de las TI: Es todo el hardware, software, redes e instalaciones, que se necesitan para desarrollar, probar, entregar, monitorear, controlar o dar soporte a servicios de las TI y a aplicaciones. El término incluye toda la TI, pero no a las personas, procesos y documentación asociados.

Nivel de Servicio: 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.

Operaciones de las TI: Son las actividades ejecutadas por el control de las operaciones de las TI, incluye la gestión de consola, puente de operaciones, programación de tareas, respaldos y restauración, gestión de impresión y de salidas. El término operaciones de las TI también se utiliza como sinónimo de operación del servicio.

Proveedor de Servicios de las TI: Es la persona que proporciona servicios a clientes internos o externos.

Servicio de las TI: Es un servicio proporcionado por un proveedor de servicios de las TI. Se compone de una combinación de las TI, personas y procesos. Los servicios de las TI de cara-al-cliente dan soporte directo a los procesos del negocio de uno o más clientes y sus objetivos de niveles de servicio deben definirse en un acuerdo de nivel de servicio.