



Facultad 1

Módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI.



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

Autor:

Roynel Mena Chávez

Tutores:

Ing. Yuneysi Barrios Pérez

Ing. Ramón Morales Alvares

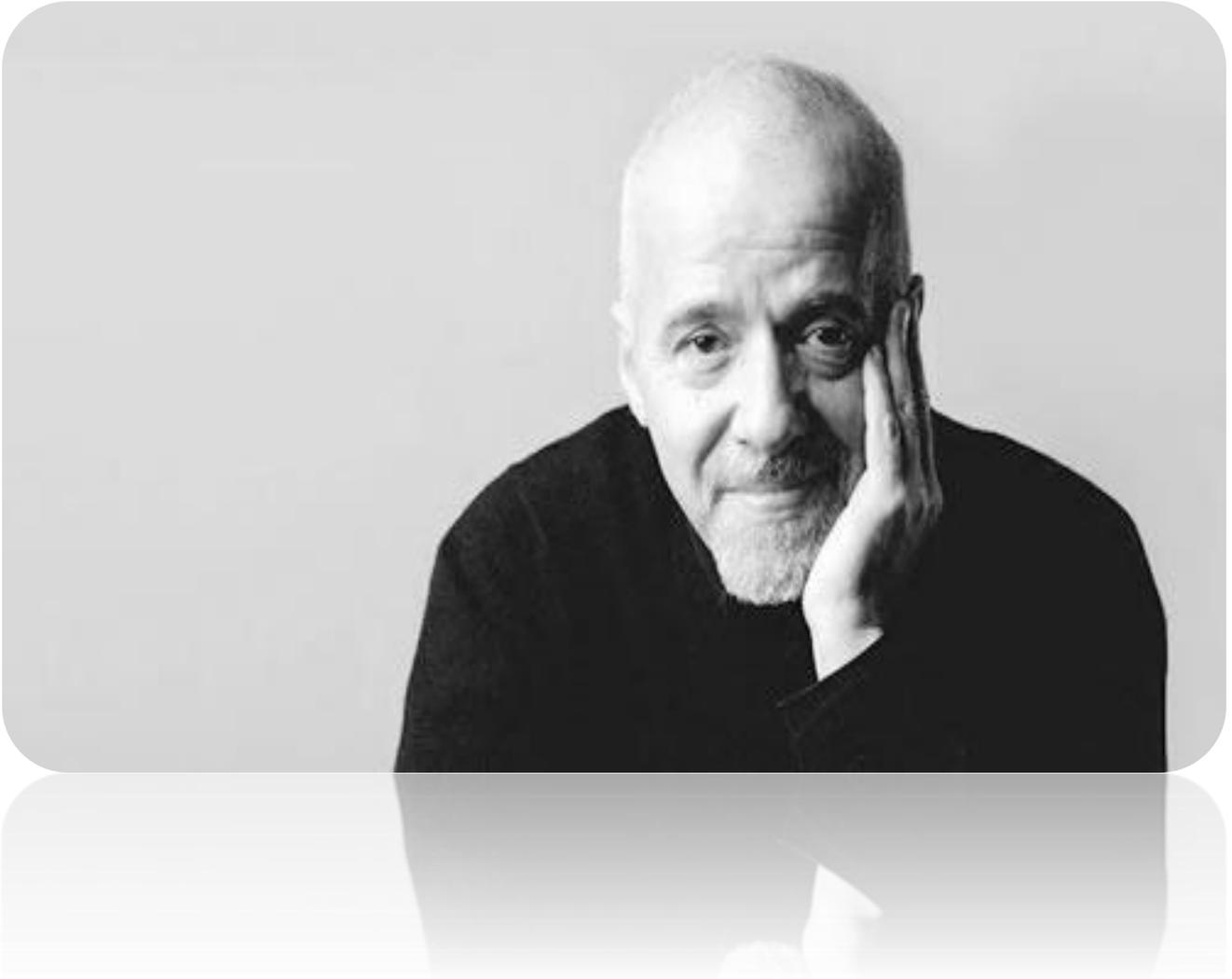
Ing. Serguey González Garay

Co-Tutor:

Ing. Nadiela Milan Cristo

La Habana, diciembre de 2019

“Año 61 de la Revolución”



Recuerda tus sueños y lucha por ellos, debes saber que quieres de la vida, ya que solo hay una cosa que hace imposible tu sueño:

el miedo al fracaso.

Paulo Coelho

Agradecimientos

Ahora sí que estoy consciente, que tengo familia, mis amigos, mis motivos que son más que suficiente, para hacer lo que más quiero porque soy valiente. . . Eso ya lo sé, ya lo sé, que cuando caiga ellos ahí estarán ¿"dispuestos para qué"? para levantarme, cuando estas piernas no me funcionen más, entonces salgo para el frente, me hice más fuerte, no quiero ser un peso para toda mi gente, pateé de repente, los problemas de siempre, le di con toda la fuerza de este ser viviente, y lo voy a hacer por mi mamá, por mi papá, por mi hermano, por mis panas, por mis ganas, por las buenas, por las malas, por la pena, por más ganas, por la risa, por el llanto, por mentiras, por engaños, por heridas, por ayer, por el hoy, porque sé, que ahora, ingeniero soy. . .

Y como de bien nacido es ser agradecido. . .

Le agradezco, primeramente, a mi colectivo de tutores, Nadiela, Yuneysi, Ramón y Serguey, gracias por ayudarme y apoyarme en esta tarea tan difícil de convertirme en ingeniero en ciencias informáticas, gracias por el tiempo que me dedicaron y sobre todo por la paciencia, especialmente a. . .

Serguey González Garay, una parte de este logro hermano, es gracias a ti, porque fuiste más que un tutor, fuiste un amigo, fuiste como un padre, preocupado, siempre atento, y yo, el niño chico, que después de las travesuras y las trastadas, siempre acudía a ti, para recibir mis respectivos alones de oreja, y también, esos buenos consejos y esas grandes y constructivas charlas, que nunca fueron en vano, a ti. . . mis más sinceros respetos y gracias, muchas gracias por todo.

Gracias también a todos los profesores y a todos los directivos de la facultad, que de una forma u otra tuvieron que ver con mi formación como futuro profesional, en esta etapa de estudiante, en estos 5 largos años.

Agradecimientos

Julio García Coste, hermano, aún recuerdo aquellas noches en las que el inglés se me hacía de rogar, y tú, con toda la paciencia del mundo, estuviste ahí para ayudar, muchas gracias por siempre estar disponible para mí, por ayudarme en todo lo que pudiste, y, sobre todo, por todo lo que me enseñaste, tanto a nivel académico como a nivel personal, ¡Muchas gracias por todo!

Gracias a todas las personas que de una manera u otra pasaron por mi vida en este tiempo, conocidos, amigos, falsos amigos, compañeras, novias y etc. porque de cada uno de ustedes aprendí, aprendí que en ocasiones las cosas no son como parecen ser, aprendí que nada es para siempre, y aprendí a diferenciar al amigo del conocido, y al conocido del amigo, aprendí que el conocido es aquel que solo da la cara cuando quiere fumar y beber, y que el amigo es aquel, que sin importarle esos términos, permanece a tu lado durante 5 años, y cumple todas las condiciones y expectativas que encierra la palabra "Amigo".

Y ahora con permiso de los conocidos y falsos amigos, voy a dar paso a los verdaderos amigos, porque sin ustedes, esta universidad no hubiese sido ni remotamente lo mismo,

Mis "SuperFriends" Darlyn, Danaysi, Eileen, Lisbet, Jorge David y Roniel, los primeros, los más entrañables, que por cosas de la vida, en un momento determinado, tristemente nos separamos, pero al final, lo que cuenta es todo lo que hicimos juntos, que es lo que realmente, a pesar de la distancia, nos une como hermanos, y pensar que un día, todos nosotros nos separaremos, extrañaremos nuestras conversaciones, pasaran días, meses y años hasta que nuestro contacto se vuelva raro, pero un día, nuestros hijos verán nuestras fotos y vídeos y nos preguntaran: ¿"Papa, quienes son estas personas?" sonreiremos, y con lágrimas en los ojos, como ahora! le diremos... "Hijos... fue con ellos que tuve los mejores años de mi vida, los de la universidad" ¡Los amo a todos!

Agradecimientos

Jessy De Los Santos López, eres y serás siempre una de las mejores personas que he conocido, en verdad no tengo palabras para expresarte todo lo que siento por ti, porque pudiste ser una persona más entre mis amigos o amigas, y sin embargo, para mi, fuiste y eres una de las personas más especiales que he tenido en mi vida, fuiste y eres más que una compañera, más que una amiga, gracias por hacerme tanto bien con tu apoyo, cariño y amor, jamás podría ni sabría como demostrarte mi agradecimiento, por todo lo que hiciste por mi. ¡Te quiero mucho Jess!

Sergio Orlando Aguilera, gracias por siempre mantenerte fiel y real a pesar de todas las cosas, por tener lo que hay que tener para decirme cuatro verdades cuando el momento lo ameritaba, por estar conmigo siempre que pudiste, por hacerme reír con tus chistes malos y tu chuchó incansable, por demostrarme siempre, a pesar de que lo sabía, que amigos, pocos, conocidos, muchos. ¡Gracias hermano! ¡Te quiero!

Alí Corrales, gracias por también ser más que un amigo, gracias por poder contar contigo siempre, y por tu hacer lo mismo conmigo también, y sobre todo, en tus momentos más duros, esos momentos en los cuales te encontrabas rodeado de hombros, y a pesar de ello, elegías llorar en el mío, así como tantas veces yo elegí hacerlo en el tuyo, porque ambos sabíamos, que a pesar de conocer la diversidad de amigos que nos rodeaba, debíamos hacerlo juntos, yo contigo y tu conmigo, gracias por todo hermano. ¡Te quiero!

Luis Ernesto...

Ana Carla, demás está decirte todo lo que por ti siento, amor, cariño, gratitud, todos esos sentimientos de amistad que es capaz de sentir una persona por una verdadera amiga, gran parte de esto es gracias a ti, sin ti, esto no hubiese sido posible, gracias por cargar conmigo siempre, por tirar de mi soga cuando estaba atrás, sin importarte lo fuerte que de ella, tuvieras

Agradecimientos

que tirar, gracias por aguantar todas mis idas y venidas y mis crisis existenciales, por escucharme, aconsejarme y mal criarme, por tenerme siempre presente y nunca descuidarme, ni dejarme atrás y ni remotamente abandonarme. A ti, un millón de gracias, te quiero Anilla.

Eric Santiesteban.....

Kilmer Hernández, hermano, que puedo decir de ti, aún recuerdo cuando nos conocimos, montando el merengue, recuerdas, y yo, todo fuera de ritmo, y ahí empezó todo, de merengue en merengue, te fuiste colando dentro de mi gente, te fuiste convirtiendo poco a poco en más amigo que los que decían serlo, con tus charlas, tus consejos y tus buenos momentos, momentos en los que me enseñaste que hay ocasiones en la vida en que uno tiene que ser una persona egoísta y dejar de pensar en los demás y pensar solo en uno, y pensar donde y con quien se siente bien en verdad, para así poder ser feliz en realidad, sin que te importe nada más. ¡Gracias por siempre estar ahí cuando te necesite, por brindarme tu apoyo y tu amistad incondicional, por dejar a un lado tu comodidad y privacidad, y acogerme en tu casa, y tratarme con un hermano, brindándome lo mejor de ti y tu apoyo incondicional, por eso y por todo lo demás, muchas gracias! ¡Te quiero Mivi!

Leo, mi Leo, mi leo leito leote... hoy es un día de esos, en los que me vienen a la cabeza tantos buenos recuerdos de la infinidad de momentos extraordinarios que hemos pasado juntos, que realmente no sabría ni por cual empezar, pero me quedo con uno, el más grande de todos, 28 de Junio de 2019, día en el cual, te encontrabas en la misma situación que yo hoy, día en el cual, te me convertiste en ingeniero, y yo, aquí estaba, pero no en tus mismas condiciones y ni remotamente a punto de estarlo, luego de haber conversado e imaginado ambos tantas veces ese momento, el cual siempre planeamos hacerlo juntos, como tantas cosas que hemos hecho a lo largo de estos cinco años, pero por cosas de la vida y también porque soy un tremendo idiota, no

Agradecimientos

pudo ser, por eso ese día lloraba y lloraba, porque sentía que, a pesar de no haberte fallado nunca, a mi entender, sentía que te había fallado, y en lo más importante, te fallé en el compromiso y en la ilusión de graduarnos juntos, pero también lloraba de alegría, de emoción, y también por que sabía, que un día como hoy, aquí yo estaría, y tu a mi lado observarías, como me convertiría, en lo mismo que yo te vi convertirte a ti: en ingeniero en ciencias informáticas. . . Quisiera decirte tantas cosas más, pero la verdad, no me alcanzaría la vida para agradecerte y decirte todo lo que significas para mí, pero creo que, con esto, ya te lo digo todo: . . .tu eres mi hermano del alma realmente el amigo, que en todo camino y jornada esta siempre conmigo, aunque eres un hombre aun tienes alma de niño, aquel que me da su amistad, su respeto y cariño, recuerdo que juntos pasamos muy duros momentos y tú no cambiaste por fuerte que fueran los vientos, es tu corazón una casa de puertas abiertas, tu eres realmente el más cierto en horas inciertas, en ciertos momentos difíciles que hay en la vida, buscamos a quien nos ayude a encontrar la salida, y aquella palabra de fuerza y de fé que me has dado, me da la certeza que siempre estuviste a mi lado, tu eres mi amigo del alma que en toda jornada, sonrisa y abrazo festivo a cada llegada, me dices verdades tan grandes con frases abiertas, tu eres realmente el más cierto en horas inciertas. . . ¡No preciso ni decir todo esto que te digo, pero es bueno así sentir que eres tu mi gran amigo! ¡Te Amo Cabrón!

Chinito. . .

Mamá y Papá, mis viejos, los amores de mi vida, las personas más importantes del mundo, a las que más amo. . .

A mis padres, por haberme forjado como la persona que soy en la actualidad, por ser los principales valedores de este gran triunfo en este largo andar como estudiante, por ser mis motores impulsores, los principales promotores de mis sueños y mis motivos de inspiración para salir adelante todos los días de estos largos años de carrera universitaria y de todos los días de mi vida, por el amor recibido, la dedicación y la paciencia con la que cada día de mi vida me educaron, por confiar y creer en mí y en mis expectativas, aunque en momentos determinados se los haya puesto difícil, por cada noche de desvelo y llanto de preocupación, por dedicarse a mí en cuerpo y alma desde mi nacimiento y hoy, les pido perdón por todos los malos momentos y tragos amargos por los cuales les he hecho pasar. Por siempre desear y anhelar lo mejor para mi vida, gracias por cada consejo, cada conversación y por cada una de sus palabras que me han guiado toda mi vida y lo seguirán haciendo, por el resto de mis días.

A mi hermano, aunque la mayoría de las veces hubiese parecido que estuviéramos en una batalla, hay momentos en que la guerra cesa y nos unimos para lograr nuestros objetivos, como grandes hermanos que somos, por siempre estar ahí y por todos los momentos que desde pequeños hemos compartido juntos...

A ustedes, especial dedicación, aquí le vas su "Título de Ingeniero en Ciencias Informáticas"

Los amo...

Declaración de autoría

Declaro por este medio que yo Roynel Mena Chávez, con carné de identidad 94041030745, soy el autor principal del trabajo titulado Módulo de Gestión Logística para el Sistema de Gestión Multieventos de la UCI, y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Firma del autor:

Roynel Mena Chávez

Firma del tutor:

Ing. Serguey González Garay

Firma del tutor:

Ing. Ramón Álvarez Morales

Firma del tutor:

Ing. Yuneisy Barrios Pérez

Firma del co-tutor:

Ing. Nadiela Milan Cristo

Resumen

La presente investigación se centra en el desarrollo de un módulo, que integrado al Sistema de Gestión Multieventos (SGM), administre la logística de los eventos que se registren en dicho sistema, de la Universidad de las Ciencias Informáticas. Se realizó un estudio de fuentes bibliográficas y sistemas web sobre la gestión logística de eventos, a nivel internacional, nacional y en el contexto de la UCI. El estudio evidenció la necesidad de desarrollar una nueva solución. Para la implementación, se empleó la metodología AUP-UCI, en su escenario 4, Drupal 7, como sistema gestor de contenido, Visual Paradigm 8.0, como herramienta para el modelado y el sistema gestor de base de datos, MySQL 5. Como resultado se obtuvo un módulo de gestión logística para el SGM. Dicho módulo permite la gestión logística de eventos mediante la realización de solicitudes de alimentación, hospedaje y transporte, para los participantes en los eventos, y solicitudes de locales y materiales, necesarios para la realización de los mismos. Brinda, además, una funcionalidad que permite exportar a pdf las solicitudes creadas y enviarlas por correo electrónico. Finalmente, el módulo se sometió a una estrategia de pruebas, compuesta por pruebas funcionales, de seguridad, rendimiento, integración y de aceptación, los errores encontrados durante las pruebas realizadas, fueron corregidos satisfactoriamente, arrojando que el módulo funciona correctamente y que sus funcionalidades no se vieron afectadas con la integración, y las pruebas de aceptación por parte del cliente y de sus usuarios finales, arrojaron la conformidad de los mismos, con el módulo implementado.

Palabras clave: gestión de logística, eventos, solicitud, módulo.

Índice de contenido

| | |
|--|--------|
| Introducción | - 1 - |
| Capítulo I: Fundamentación teórica | - 6 - |
| 1.1. Marco conceptual | - 6 - |
| 1.2. Análisis de sistemas homólogos | - 7 - |
| 1.2.2. Sistemas Homólogos Nacionales | - 9 - |
| 1.2.3. Conclusiones del análisis de sistemas homólogos | - 10 - |
| 1.3. Tecnologías, herramientas y metodologías asociadas al desarrollo de la solución | - 11 - |
| 1.3.1. Sistemas de gestión de contenido | - 12 - |
| 1.3.2. Metodología para el desarrollo de software | - 14 - |
| 1.3.3. Lenguaje de desarrollo | - 15 - |
| 1.3.4. Entorno integrado de desarrollo | - 17 - |
| 1.3.5. Herramienta de modelado | - 18 - |
| 1.3.6. Sistema gestor de bases de datos | - 19 - |
| 1.3.7. Servidor web | - 19 - |
| 1.3.8. Herramientas de pruebas | - 20 - |
| 1.4. Conclusiones del capítulo | - 21 - |
| Capítulo II: Características y arquitectura del módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI | - 22 - |
| 2.1. Descripción de la propuesta de solución | - 22 - |
| 2.2. Requisitos de software | - 22 - |
| 2.2.1. Requisitos Funcionales | - 23 - |
| 2.2.2. Requisitos no funcionales | - 25 - |
| 2.3. Historias de usuario | - 27 - |
| 2.4. Arquitectura | - 31 - |
| 2.4.1. N-Capas de Drupal | - 31 - |
| 2.5. Patrones de diseño | - 32 - |

| | |
|---|---------------|
| 2.5.1. Patrones GoF..... | - 33 - |
| 2.6. Modelo de despliegue | - 34 - |
| 2.6.1. Descripción de los elementos del modelo de despliegue | - 34 - |
| 2.7. Modelo de Datos | - 35 - |
| 2.8. Conclusiones del capítulo | - 36 - |
| Capítulo III: Implementación y validación del módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI..... | - 37 - |
| 3.1. Diagrama de componentes..... | - 37 - |
| 3.2. Estándares de codificación..... | - 39 - |
| 3.2.1. Indentación y espacios en blanco | - 39 - |
| 3.2.2. Operadores..... | - 40 - |
| 3.2.3. Estructuras de control | - 40 - |
| 3.2.4. Longitud de líneas..... | - 40 - |
| 3.2.5. Arreglos | - 41 - |
| 3.2.6. Uso de comillas..... | - 41 - |
| 3.2.7. Etiquetas de apertura y cierre de PHP | - 42 - |
| 3.2.8. Uso de punto y coma (;) en código PHP | - 42 - |
| 3.2.9. Nombre de funciones | - 42 - |
| 3.2.10. Nombre de variables | - 42 - |
| 3.3. Validación de la propuesta de solución..... | - 43 - |
| 3.3.1. Pruebas funcionales..... | - 43 - |
| 3.3.2. Pruebas de seguridad | - 45 - |
| 3.3.3. Pruebas de rendimiento | - 46 - |
| 3.3.4. Pruebas de integración | - 48 - |
| 3.3.5 Pruebas de aceptación | - 49 - |
| 3.4 Análisis de la aplicación de la técnica de ladov | - 50 - |
| 3.4.1. Resultados obtenidos de la aplicación de la técnica de ladov | - 51 - |

| | |
|---|--------|
| 3.5. Conclusiones del capítulo..... | - 53 - |
| Conclusiones | - 54 - |
| Recomendaciones | - 56 - |
| Referencias bibliográficas | - 57 - |
| Anexos | - 61 - |

Índice de figuras

| | |
|--|--------|
| Figura 1. Fases e Iteraciones de la variante de AUP para la UCI..... | - 15 - |
| Figura 2. Arquitectura de la aplicación | - 32 - |
| Figura 3. Modelo de Despliegue | - 34 - |
| Figura 4. Modelo de datos | - 36 - |
| Figura 5. Diagrama de componentes | - 39 - |
| Figura 6. Uso de indentación y espacios en blanc. | - 40 - |
| Figura 7. Uso de operadores. | - 40 - |
| Figura 8. Uso de estructuras de control | - 40 - |
| Figura 9. Longitud de línea con menos de 80 caracteres | - 41 - |
| Figura 10. Uso de arreglos..... | - 41 - |
| Figura 11. Uso de comillas..... | - 42 - |
| Figura 12. Uso de etiquetas PHP | - 42 - |
| Figura 13. Uso de punto y coma | - 42 - |
| Figura 14. Nombre de funciones. | - 42 - |
| Figura 15. Nombre de variables | - 43 - |
| Figura 16. Comportamiento de las no conformidades por iteración..... | - 45 - |
| Figura 17. Resultado de la prueba de rendimiento..... | - 48 - |
| Figura 18. Interfaz gráfica Solicitud de Materiales..... | - 49 - |

Índice de tablas

| | |
|--|--------|
| Tabla 1. Comparación entre las aplicaciones estudiadas | - 10 - |
| Tabla 2: Requisitos Funcionales | - 23 - |
| Tabla 3. Resumen de la etapa de captura de los RF | - 25 - |
| Tabla 4. HU_ Eliminar solicitud de locales. | - 28 - |
| Tabla 5. HU_ Crear solicitud de transporte | - 30 - |
| Tabla 6. Caso de prueba 1..... | - 43 - |
| Tabla 7. Resultados de las pruebas de seguridad..... | - 46 - |
| Tabla 8. Cuadro Lógico de ladov utilizado | - 50 - |
| Tabla 9. Escala del Índice de Satisfacción Individual | - 51 - |

Introducción

El perfeccionamiento constante del conocimiento adquirido ha devenido en el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC). Las TIC se definen como un conjunto de herramientas y procesos para acceder, recuperar, guardar, organizar, manipular, producir, intercambiar y presentar información por medios electrónicos. Esto incluye hardware, software y telecomunicaciones en la forma de computadores y programas tales como aplicaciones multimedia y sistemas de bases de datos (González, y otros, 2015).

La sociedad científica moderna no escapa del espectro de oportunidades que hoy ofrecen las TIC. En la actualidad son grandes los desafíos a los que se enfrenta el conocimiento científico. La gran mayoría de los eventos académicos y científicos actualmente son soportados por sistemas informáticos que colaboran en la socialización del conocimiento. La comunidad científica cubana también aprovecha el uso de las TIC y numerosos son los esfuerzos que realizan las universidades cubanas en materia de ciencia, entre ellas, la Universidad de las Ciencias Informáticas (UCI). Desde su creación, la UCI ha tenido como uno de sus objetivos principales, el desarrollo de soluciones informáticas que respondan a las necesidades de informatización de la sociedad y hagan frente a las limitaciones tecnológicas que afectan al país. Los resultados alcanzados en materia de ciencia se corresponden con la labor de investigación científica que hoy se realizan en esta casa de altos estudios, que se pueden difundir a través de diversos espacios, entre los cuales se pueden encontrar los eventos académicos y científicos.

La Universidad de las Ciencias Informáticas organiza un gran número de eventos académicos y científicos, de naturaleza nacional, regional e internacional. Entre sus objetivos se encuentra: incrementar los resultados de la investigación, desarrollo y la gestión de la innovación, de manera que desempeñen un papel decisivo en el desarrollo tecnológico, económico y social del país.

La gestión de los eventos es manejada por el Sistema de Gestión Multieventos. Esta plataforma electrónica es un sistema multisitio encargado de gestionar eventos, específicamente los desarrollados en la UCI, dentro de sus capacidades no contempla la gestión logística que demandan dichos eventos, la cual está comprendida por el manejo de alimentación, hospedaje y transporte para los participantes en los eventos, y de los locales y materiales necesarios para la realización de los mismos. Con esta aspiración, la Dirección

Módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI.



de Ciencia, Tecnología e Innovación, se enfrenta a varios contratiempos que dificultan en alguna medida la adecuada organización de los eventos.

Algunos de estos contratiempos se enuncian a continuación:

En la universidad, la documentación formal que existe para la gestión logística de eventos, se encuentra en formato duro, no digitalizada, lo que, en algunos casos, afecta la buena planificación y ejecución de este tipo de actividades.

La gestión logística se realiza de forma manual, propiciando un mayor consumo de tiempo a la hora de ser procesadas las solicitudes que conforman esta gestión logística para los eventos académicos y científicos de la UCI.

Al realizarse de manera manual esta gestión, puede existir posibles pérdidas de información o duplicación de la misma, impidiendo el control adecuado de cada uno de los procesos que requiere esta gestión.

La ejecución de los eventos demanda muchos recursos humanos y materiales, y es evidente la falta de esta digitalización para la organización del evento, bajo el enfoque de esta gestión logística.

Para lograr el mayor porcentaje de éxito de los eventos, se requiere de la participación e interacción de diversas áreas de la universidad como Logística, Contabilidad y Finanzas, Economía, Tecnología, Transporte, Residencia, entre otras, y con la forma manual actual de manejar la logística de los eventos, no se garantiza una interacción rápida y eficiente de todas ellas, puesto que, una vez creadas las solicitudes, la entrega de estas, se realiza personalmente a los diferentes responsables de aprobarlas por las distintas áreas, y una vez aprobadas, se realiza el mismo proceso de igual manera a la inversa, constituyendo un consumo de tiempo importante.

Teniendo en cuenta la situación planteada anteriormente, se define el siguiente **problema de la investigación**: ¿Cómo contribuir a la mejora de la gestión logística de los eventos académicos y científicos en el Sistema de Gestión Multieventos en la Universidad de las Ciencias Informáticas?

Se define como **objeto de estudio**: el proceso de gestión logística de eventos académicos y científicos, el cual se enmarca al siguiente **campo de acción**: en el proceso de gestión logística de los eventos académicos y científicos de la UCI.

Módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI.



Para solucionar el problema de investigación antes mencionado, se plantea como **objetivo general**:

Desarrollar un módulo que contribuya a la mejora de la gestión logística de los eventos académicos y científicos en el Sistema de Gestión Multieventos de la UCI.

Para guiar el proceso investigativo se plantean las siguientes **preguntas científicas**:

- ¿Cuáles son los fundamentos teóricos que sustentan el proceso de análisis de la gestión logística de eventos?
- ¿Cuáles son las tecnologías adecuadas para la implementación del módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI?
- ¿Qué elementos de la metodología de desarrollo de software se deben tener en cuenta en el diseño del módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI.?
- ¿Cómo materializar en términos de componentes el módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI?
- ¿Cómo evaluar el funcionamiento del módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI?

Con el propósito de dar cumplimiento a las preguntas científicas planteadas anteriormente se definen las siguientes **tareas de investigación**:

- Análisis de los fundamentos teóricos que sustentan el proceso de gestión logística de eventos académicos y científicos.
- Valoración del estado del arte de las herramientas o aplicaciones que realicen análisis de gestión de la logística de eventos académicos y científicos.
- Definición de la metodología a utilizar en el desarrollo Módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI.
- Implementación del Módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI.
- Valoración de los resultados de las pruebas al módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI.



Métodos de investigación científica

Para el desarrollo de la presente investigación resulta primordial realizar un correcto uso de los **métodos de investigación científica**, que permitan la obtención de la información necesaria referente al objeto de estudio. A continuación, se muestran los métodos científicos utilizados para el desarrollo de esta investigación:

Métodos teóricos: Permiten estudiar las características del objeto de investigación que no son observables directamente y posibilitan el conocimiento del estado del arte del fenómeno, su evolución en una etapa determinada, su relación con otros fenómenos, así como su aislamiento como objeto estudiado (Cabrera Campos, 2012).

- **Análisis Histórico-Lógico:** Permite estudiar la evolución de los procesos de gestión logística de eventos académicos y científicos en instituciones cubanas y extranjeras, para su eficiente administración, así como los diferentes sistemas gestores utilizados por ellos. Facilita el análisis de las soluciones informáticas existentes en la actualidad que aborden el tema a tratar.
- **Analítico-Sintético:** Permite el estudio de los procesos de gestión logística de eventos académicos y científicos en las entidades, mediante la revisión de documentos y sitios web relacionados con el tema a abordar. Facilitó el análisis de las herramientas y tecnologías a utilizar en el sistema.

Métodos Empíricos:

Describen y explican las características fenomenológicas del objeto. Representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional. Son los llamados métodos de recolección de información en un proceso investigativo, de los cuales se pueden fijar: la observación, la entrevista, el cuestionario y el análisis documental (Cabrera Campos, 2012).

- **Entrevista:** Este método es de gran utilidad para recopilar información cualitativa, a través de la cual se pudo precisar el problema de investigación y las necesidades del cliente. La aplicación de este método posibilitó entender cómo se realiza actualmente el proceso de gestión logística de eventos académicos y científicos en la UCI, permitiendo identificar los requisitos funcionales y no funcionales que debe cumplir el módulo como propuesta de solución.

El presente trabajo está estructurado en 3 capítulos, distribuidos de la siguiente forma:

Módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI.



Capítulo I. Fundamentación teórica del módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI: En este capítulo se aborda la fundamentación teórica, que sirve de base a la investigación del problema planteado, además se lleva a cabo un estudio del estado del arte del tema en cuestión, a nivel nacional e internacional, que ayuda a la selección de las herramientas existentes, referentes a la situación identificada, y se describen las tecnologías y metodologías en las que se apoya la solución del problema.

Capítulo II. Características y arquitectura del módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI: En este capítulo se expone la propuesta de solución de manera detallada, describiéndose los requisitos funcionales y no funcionales, así como todo lo referente a la arquitectura del módulo y a los patrones de diseño utilizados.

Capítulo III: Implementación y validación del módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI: En este capítulo se detalla la construcción y validación de la propuesta de solución al problema planteado. Se realizan además y describen las pruebas de software a la aplicación.

El presente trabajo contiene además **Conclusiones, Recomendaciones, Referencias bibliográficas, Bibliografía consultada, Glosario de términos, Acrónimos y Anexos.**

Capítulo I: Fundamentación teórica

Introducción

En el siguiente capítulo se abordan los principales conceptos que están relacionados con los procesos que integran el campo de acción, se hace un estudio del estado del arte relacionado con soluciones informáticas existentes para la gestión de eventos académicos y científicos, y se describen las tecnologías, herramientas y la metodología asociadas al desarrollo de la solución.

1.1. Marco conceptual

Para una mejor comprensión del contexto de la investigación, en esta sección se enuncian algunos conceptos fundamentales que contribuyen a ello.

Evento: Suceso importante y programado, de índole social, académica, científica, artística o deportiva (Real Academia Española, 2016).

Evento Académico: Según el Consejo de Desarrollo Científico y Humanístico de la Universidad Central de Venezuela se consideran eventos académicos al conjunto de actividades de carácter profesional, artístico o técnico que implica algún tipo de enseñanza o capacitación. Este tipo de eventos se realizan con la finalidad de que profesionales de distintas áreas, los aprovechen y enriquezcan su formación integral (Nicolás, 2015).

Evento científico: Según el Consejo de Desarrollo Científico y Humanístico de la Universidad Central de Venezuela se consideran eventos científicos las reuniones nacionales e internacionales tales como: congresos, seminarios, conferencias, foros, jornadas y simposios destinados a la divulgación e intercambio de resultados originales de investigación (Nicolás, 2015).

Un evento científico es una actividad para establecer un mecanismo que ayude a la generación, fortalecimiento y consolidación del conocimiento científico en los diferentes sectores de la entidad. Es un encuentro donde se estimula el conocimiento adquirido y se presenta información acerca de un tema específico. De esta manera, los investigadores enriquecen su formación profesional y aumentan su capacidad investigativa a través del intercambio entre ellos y con los desarrolladores (Cardoso Said, y otros, 2013).

Gestión: Es la administración, organización y funcionamiento de una empresa, actividad económica u organismo (Real Academia Española, 2016).

Logística: Conjunto de los medios necesarios para llevar a cabo un fin determinado de un proceso complicado (Real Academia Española, 2016).

Gestión de Logística: Se puede definir la gestión de la logística como la gestión del flujo de materias primas, productos, servicios e información a lo largo de toda la cadena de suministro de un producto o servicio. En una definición informal se puede entender la logística como la gestión de todas las operaciones que buscan garantizar la disponibilidad de un determinado elemento (producto, servicio, información) en tiempo y forma óptimos (Real Academia Española, 2016).

Módulo: En programación, un módulo es una porción de un programa de ordenador. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente, una de dichas tareas (o varias, en algún caso) (Drupal. 2016.).

1.2. Análisis de sistemas homólogos

Con el objetivo de darle visibilidad a los eventos, se han desarrollado disímiles sistemas que promocionan y gestionan logística de eventos académicos y científicos. Por consiguiente, para desarrollar un nuevo sistema se hace necesario estudiar los antecedentes del mismo. A continuación, se muestra un estudio de una serie de sistemas que guardan relación con la presente investigación, puesto que, en este sentido, no han sido pocas las instituciones a nivel nacional e internacional, que han realizado grandes esfuerzos con el fin de informatizar dichos eventos.

1.2.1. Sistemas Homólogos Internacionales

1.2.1.1. Open Conference Systems

Open Conference Systems (por sus siglas en inglés OCS) es un producto de la universidad de Columbia Británica, que está desarrollado por el proyecto *Public Knowledge Project*¹ (por sus siglas en inglés, PKP) actualmente en su versión 2.3.6. Constituye una herramienta de publicación web gratuita que provee apoyo tecnológico a la organización de conferencias o eventos en general.

¹ Se dedica a mejorar la calidad académica y pública de la investigación. Proporciona el software de código abierto para publicación y gestión de conferencias y publicaciones.



El OCS permite la creación de sitios web de una o varias conferencias simultáneas y en varios idiomas, así como envíos de trabajos. Además, permite la redacción y envío de convocatorias de ponencias, ya sea por anuncios en el sitio o envío de correos masivos mediante la plataforma. También posibilita monitorear y controlar el proceso de revisión de los trabajos; recibir y aceptar, en línea, las propuestas de ponencias y su resumen, así como su exportación a pdf. Asimismo, brinda la posibilidad de publicar los trabajos aprobados e incluso manejar comentarios o discusiones pasado el evento. Realizar búsquedas de trabajos por diferentes criterios, dígame: nombre, autor y tema, efectuar registros y pago en línea. Permite generar informes y estadísticas del evento: número de inscritos, trabajos aprobados y rechazados, listado de participantes, ponentes e integrar debates en línea después de una conferencia (Brenji, 2012).

OCS es un producto de código abierto² que está bajo la licencia de GNU General Public License³ (GNU GPL). Su desarrollo completamente en lenguaje de programación PHP y puede utilizar base de datos MySQL o Postgres SQL (Fraser, 2014). Además, es el sistema estudiado presenta una interfaz de usuario poco clara y compleja, puesto que está diseñado para la organización de eventos de los cuales se maneja numerosa información (Brenji, 2012).

1.2.1.2. EasyChair

EasyChair es un sistema de gestión de conferencias que es flexible, fácil de usar y tiene muchas características para que sea adecuado para varios modelos de conferencias. Actualmente es probablemente el sistema de gestión de conferencias más utilizado. Sus usuarios incluyen conferencias principales en varias áreas, por ejemplo, en World Wide Web y en Bioinformática. También, sus usuarios van desde pequeños talleres con alrededor de 10 presentaciones hasta grandes conferencias con miles de presentaciones.

La primera versión de EasyChair se implementó en 2002. Fue utilizada por 12 conferencias en 2002-2004 y por 66 conferencias en 2005. Desde 2006, se ha convertido en el sistema de gestión de conferencias número uno en número de conferencias, usuarios y presentaciones.

²Código abierto: en inglés, Open Source. Es el término con el que se conoce al software distribuido y desarrollado libremente.

³ GNU GPL, en español Licencia Pública General de GNU, creada originalmente por Richard Stallman, la cual garantiza la libertad de compartir y modificar todas las versiones de un programa.

EasyChair fue diseñado para ayudar a los organizadores de la conferencia a enfrentar la complejidad del proceso de arbitraje. La versión actual soporta gestión y seguimiento del comité del programa; gestión sofisticada y flexible del acceso de los miembros de PC y los árbitros a documentos y conflictos de intereses; presentación automática en papel; asignación de papel basada en las preferencias de los miembros de PC; lista de los últimos eventos; presentación de opiniones enviando correos electrónicos a miembros de PC, árbitros y autores; seguimiento de correo electrónico; discusión en línea de artículos; la fase de respuesta del autor (también conocida como refutación), cuando el autor puede responder a las revisiones; preparación de actas de congresos, generación y publicación del programa de conferencias, generación de folletos de conferencias con programa, Programa de conferencia y procedimientos en unidades flash y muchas otras características.

Actualmente, EasyChair soporta dos modelos. El modelo estándar está destinado a conferencias que tengan un solo comité de programa. Los documentos se distribuyen entre los miembros de PC, normalmente en función de sus preferencias. La versión multipista es para conferencias que tienen múltiples pistas. Cada pista tiene su propio comité de programa y una o más sillas de pista. Este modelo requiere una "silla" para supervisar o supervisar el trabajo de las pistas. Para cada uno de los modelos compatibles, los usuarios pueden elegir entre varias opciones para ajustar el sistema a sus requisitos. Por ejemplo, pueden optar por utilizar envíos anónimos. EasyChair está disponible con varias licencias, incluida una gratuita.

1.2.2. Sistemas Homólogos Nacionales

1.2.2.1. Sistema de gestión para el control de actividades científicas de los docentes

El Sistema de Gestión para el Control de Actividades Científicas de los Docentes (SGCACD), desarrollado en la Facultad de Ciencias Técnicas (FCT) perteneciente a la Universidad de Las Tunas (ULT) Vladimir Ilich Lenin, permite el control y seguimiento de las actividades científicas que se desarrollan en los departamentos docentes de la FCT. El SGCACD fue diseñado utilizando el marco de trabajo CodeIgniter y el lenguaje de programación PHP, como gestor de base de datos utilizan Postgres SQL y como metodologías de desarrollo Rational Unified Process⁴ (RUP) con modelado ágil (Rosabal, 2011).

⁴ Traducción del español: Proceso Unificado de Rational. Es un proceso de desarrollo de software que define quién está haciendo qué, cuándo y cómo alcanzar un determinado objetivo.



El sistema desarrollado permite gestionar las actividades científicas, actualizar estados de publicación y elaboración de trabajos, actualizar tipos de trabajo y tipos de documentos, registrar premios obtenidos, asignar premios y actividades. Además, permite realizar búsquedas de trabajos por diferentes criterios y adjuntar el resumen de los trabajos a presentar, gestionar locales para la realización de las actividades científicas, así como, los materiales y los medios tecnológicos necesarios para la realización de las mismas. Actualmente, este sistema se encuentra en desarrollo de nuevas versiones (Rosabal, 2011).

1.2.2.2. Plataforma electrónica para la gestión de eventos científicos en la UCI

Esta plataforma electrónica es un sistema multisitio, encargada de gestionar los eventos científicos, específicamente los desarrollados en la UCI. Posee una interfaz visualmente atractiva y es de fácil configuración. Diseñado específicamente para optimizar el tiempo y el esfuerzo en el proceso de gestión de eventos científicos.

El proceso de desarrollo fue guiado por la metodología ágil XP. Para el desarrollo del sistema se utilizaron herramientas como CMS Drupal en su versión 7.12 y MySQL en su versión 5.2.17, siendo este el gestor de base de datos que recomienda Drupal (Pérez Reyes, y otros, 2012). Este sistema permite: gestionar usuario, encuestas, ponencias, convocatorias, talleres, roles y permisos. Además, el envío de notificaciones automáticas vía correo electrónico y hacer búsquedas generales.

1.2.3. Conclusiones del análisis de sistemas homólogos

A continuación, en la **tabla 1**, se expone la comparación entre las aplicaciones estudiadas:

Tabla 1. Comparación entre las aplicaciones estudiadas (Elaboración propia).

| Sistema/Funcionalidad | Gestión de alimentación | Gestión de hospedaje | Gestión de locales | Gestión de materiales | Gestión de transporte | Envío de correos | Exportar a PDF |
|------------------------------|--------------------------------|-----------------------------|---------------------------|------------------------------|------------------------------|-------------------------|-----------------------|
| OCS | No | No | Si | No | No | Si | Si |
| EasyChair | No | No | Si | No | No | Si | Si |
| SGCADC | No | No | Si | Si | No | Si | Si |
| Plataforma UCI | No | No | Si | No | No | Si | Si |

El estudio realizado evidencia que algunos de los sistemas anteriormente mencionados no gestionan la logística necesaria para satisfacer la gestión logística que requiere la presente investigación, en el caso de los sistemas estudiados a nivel internacional, el OCS, la logística que gestiona es la publicación de los



trabajos aprobados, pago en línea, generar informes y estadísticas del evento: número de inscritos, trabajos aprobados y rechazados, así como un listado de la cantidad de participantes y ponentes, y los locales utilizados para llevar a cabo las conferencias; en el caso de EasyChair, la logística que comprende, se centra en la cantidad de conferencias que gestiona, los locales para realizarlas, el acceso de los miembros de PC y los árbitros a documentos y conflictos de intereses, listado de los últimos eventos, y la cantidad de árbitros y autores. En el caso de los sistemas estudiados a nivel nacional, el Sistema de Gestión para el Control de Actividades Científicas de los Docentes (SGCACD), desarrollado en la Facultad de Ciencias Técnicas (FCT) perteneciente a la Universidad de Las Tunas (ULT) Vladimir Ilich Lenin, la parte logística que gestiona se basa en el registro de la cantidad de premios obtenidos, la asignación de premios y la cantidad de actividades a llevarse a cabo, así como la gestión de locales para la realización de las actividades científicas, los materiales y los medios tecnológicos necesarios para la realización de las mismas. En el contexto de la UCI, la Plataforma Electrónica para la Gestión de Eventos Científicos, solo contempla la gestión de los locales en los que se realizan los eventos científicos. Lo explicado anteriormente propicia la necesidad de desarrollar una nueva propuesta de solución que integrada al Sistema de Gestión Multieventos, permita la gestión logística de los eventos que se administran en dicho sistema. No obstante, estos sistemas mencionados poseen funcionalidades en común que pueden ser utilizados en la propuesta de solución, como son: exportar a pdf y el envío de correos electrónicos mediante la plataforma, la gestión de locales para la realización de las actividades científicas, los materiales y los medios tecnológicos necesarios para la realización de las mismas.

1.3. Tecnologías, herramientas y metodologías asociadas al desarrollo de la solución

Para el desarrollo de la propuesta de solución, se hizo necesario llevar a cabo un análisis de las características particulares de un conjunto de herramientas, así como de las ventajas que reportaría el uso de cada una de ellas, ya que estos son elementos importantes a tener en cuenta a la hora de seleccionar las herramientas a utilizar, para dar solución a la investigación. Para ello se utilizan un conjunto de tecnologías y herramientas establecidas, que deben ser utilizadas en el desarrollo de sistemas de información basados en tecnologías web. Estas cumplen además con las políticas de software libre del país para lograr la soberanía tecnológica.

1.3.1. Sistemas de gestión de contenido

En los últimos años se ha desarrollado considerablemente el concepto de Sistema de Gestión de Contenidos (Content Management System y por sus siglas en inglés, CMS), es un software que se utiliza principalmente para la creación y gestión de sitios web ya sea en internet como en una intranet. Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores proporcionando un entorno que posibilita la actualización, mantenimiento y ampliación del portal web con la colaboración de múltiples usuarios (Paulsen, 2011).

Existen CMS de código abierto y de código privativo. Los CMS de código abierto brindan más libertad a los desarrolladores de portales web en cuanto a modificación y adaptabilidad con el fin de lograr la funcionalidad deseada. Entre estos CMS de código abierto los más utilizados a nivel mundial son Joomla, WordPress y Drupal (Paulsen, 2011).

Para la realización de la propuesta de solución se utilizará como CMS de código abierto Drupal en su versión 7, puesto que, la propuesta de solución en cuestión, es un módulo que va a ser integrado al Sistema de Gestión Multieventos de la UCI, el cual utiliza Drupal 7 como CMS.

1.3.1.1. Drupal 7

Drupal es un CMS de código abierto, altamente modular con énfasis en la colaboración. Contiene funcionalidades básicas en su núcleo y se puede agregar funcionalidades a través de la instalación de módulos. También separa con éxito la gestión de contenido de la presentación de contenidos. Es distribuido bajo los términos de la Licencia Pública General (GNU/GPL). Además de proporcionar herramientas para la construcción de sitios web, ofrece vías para que programadores y desarrolladores personalicen Drupal usando módulos (Drupal, 2016).

El software está desarrollado con el lenguaje de programación PHP, utiliza una base de datos MySQL y está maquetado con hojas de estilo CSS. Drupal permite implementar un blog personal o profesional, portales corporativos, tiendas virtuales, redes sociales y comunidades virtuales (Drupal, 2016).

Módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI.



Al ser la propuesta de solución, un módulo que va a ser integrado al Sistema de Gestión Multieventos, se elige Drupal 7 como sistema gestor de contenido, puesto que esta es la metodología usada por dicho sistema, anteriormente mencionado.

A continuación, se exponen una serie de características y ventajas de Drupal 7, como CMS:

Características de Drupal:

- Sistema de módulos flexible: su potente núcleo se puede extender fácilmente a través de módulos personalizados
- Sistema de temas personalizable: Las salidas en Drupal son totalmente personalizables.
- Sistema de contenido extensible y de entidades: Se pueden definir nuevos tipos de contenido (*blogs*, eventos, noticias) y agregar campos para distintos tipos de contenido.
- Permisos de acceso basado en Roles: Cada usuario en Drupal contiene un rol con privilegios determinados.

Ventajas de Drupal:

Drupal es un *software* de código abierto con una comunidad de más de 648000 usuarios y más de 10000 desarrolladores.

- Soporta más de 55 idiomas.
- El sitio oficial de Drupal lista alrededor de 12101 módulos y temas gratuitos para extender sus funcionalidades.
- Cientos de universidades del mundo utilizan Drupal para el desarrollo de sitios web.
- Algunos de los sitios más potentes del mundo utilizan esta poderosa, flexible y escalable tecnología.

Algunos avances de Drupal 7 respecto a su versión anterior están relacionadas con seguridad, usabilidad y rendimiento. Los mismos se mencionan a continuación:

Seguridad:

- Implementación segura para tareas programadas.
- Contraseña del sistema segura.
- Autenticación segura del sistema.
- Los módulos pueden ser actualizados vía web.

Usabilidad:

- Los enlaces administrativos para editar los elementos de las páginas existentes están disponibles en todas las páginas web, sin tener que ir a una página de administración primero.
- Posee un validador de fortaleza de contraseña para que sea más segura.
- Presenta pestañas verticales, un componente de interfaz reutilizable que ofrece resúmenes automáticos y aumenta la facilidad de uso.
- Presenta soporte para usos de horario.

Rendimiento:

- Incluye un campo específicamente para la carga de archivos, utilizando el módulo *FileField*.
- Incluye un campo específicamente para subir imágenes, utilizando el módulo *ImageField*.
- Campos de datos personalizados se pueden unir a los nodos, usuarios, comentarios y términos de la taxonomía.

1.3.2. Metodología para el desarrollo de software

Una metodología de desarrollo de software describe un entorno que es usado para organizar, planificar, y dirigir un proceso de desarrollo de un *software*. Existen varias metodologías de desarrollo, todas contienen algunas etapas básicas del ciclo de desarrollo como son la planificación, análisis, diseño, implementación y mantenimiento (Martinez, 2016).

1.3.2.1. Metodología AUP – UCI

Para guiar el proceso de desarrollo de la presente investigación, se seleccionó la metodología AUP, en su variación UCI, ya que es la metodología que utiliza el Sistema de Gestión Multieventos, plataforma a la cual va a ser integrado el módulo de la propuesta de solución, y, además, es la metodología definida por el proyecto. Se decidió optar por el escenario 4, el cual modela el sistema mediante historias de usuario y es conveniente en proyectos no muy extensos. De esta forma se logra estandarizar el proceso de desarrollo del software, dando cumplimiento a las buenas prácticas que define la Integración del Modelo de Madurez de Capacidades (CMMI-DEV) (Rodríguez Sánchez, 2015).

La misma cuenta con tres fases descritas a continuación:

- **Inicio:** durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información



fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo, para decidir si se ejecuta o no el proyecto.

- **Ejecución:** en esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
- **Cierre:** en esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

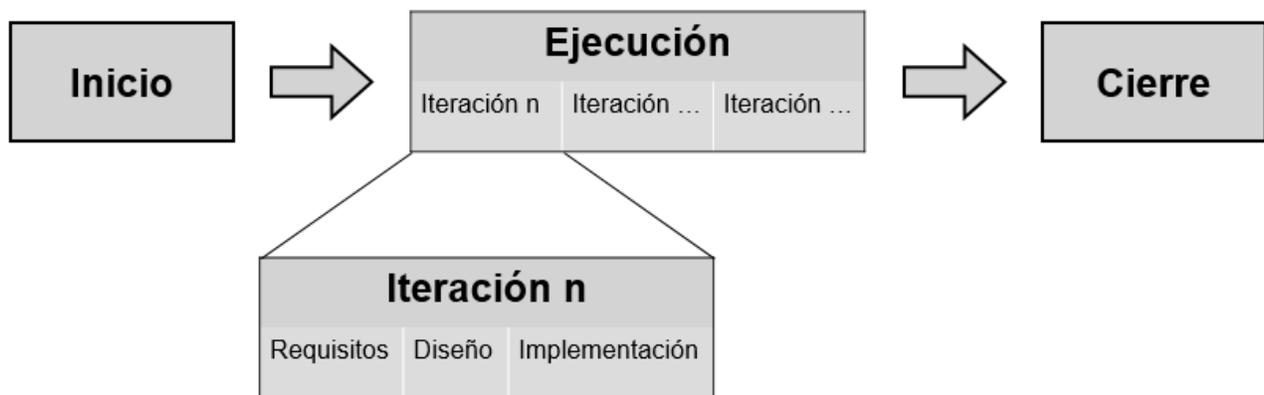


Figura 1. Fases e Iteraciones de la variante de AUP para la UCI (Rodríguez Sánchez, 2015).

En una iteración se repite el flujo de trabajo de las disciplinas, Requisitos, Análisis y diseño, Implementación y Pruebas internas. De esta forma se brinda un resultado más completo para un producto final de manera creciente. Para llegar a lograr esto, cada requisito debe tener un completo desarrollo en una única iteración (Rodríguez Sánchez, 2015).

1.3.3. Lenguaje de desarrollo

Como propuesta de solución a la situación problemática de esta investigación se tiene el desarrollo de un módulo en Drupal 7, por lo que es necesaria la selección de lenguajes de programación y de un lenguaje de modelado. En el presente acápite se caracterizan los lenguajes seleccionados para el desarrollo del módulo web.



Para el desarrollo de aplicaciones web existen gran variedad de lenguajes de programación (ASP, ActionScript, Perl, Ajax entre otros). El uso del CMS Drupal restringe el conjunto de lenguajes de programación por lo que se utilizará PHP. Para la creación de los diagramas de diseño e implementación del portal, se utilizará el lenguaje de modelado unificado (UML), por ser un lenguaje estándar en el desarrollo de software profesional. Se hace necesario además utilizar las Hojas de Estilo en Cascada (Cascading Style Sheets y por sus siglas en inglés CSS) para complementar la apariencia, funcionalidad y desarrollo del portal, así como el lenguaje de marca de hipertexto (HTML), para la correcta funcionalidad del módulo.

1.3.3.1. PHP

El pre-procesamiento de hipertexto (PHP) es un lenguaje de código abierto muy popular, especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. El código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, desconociendo así el código subyacente que era antes de la ejecución del mismo. El servidor web puede ser incluso configurado para que procese todos los ficheros HTML con PHP. Aunque el desarrollo de PHP está centrado en la programación de scripts del lado del servidor, se puede utilizar para el desarrollo de otras aplicaciones. PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) entre otros, tiene soporte para el intercambio de datos complejos entre los lenguajes de programación web (Enamorado, 2014).

1.3.3.2. UML 5

El lenguaje de modelado unificado (UML) es un lenguaje gráfico para visualizar, documentar y especificar cada una de las partes que comprende el desarrollo de software. Es un lenguaje de modelado formado por símbolos que es utilizado por muchas metodologías. Mediante el lenguaje UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código (Rodríguez, 2010).

1.3.3.3. CSS 3

CSS es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre el estilo y el formato de sus documentos. CSS se utiliza para dar estilo a documentos HTML y XML,

separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento (Cascading Style Sheets, 2013).

1.3.3.4. HTML 5

HTML 5 provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado oficialmente, pero, incluso cuando algunas API (interface de programación de aplicaciones) y la especificación de CSS3 por completo no son parte del mismo. HTML 5 es considerado el producto de la combinación de HTML, CSS y JavaScript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML 5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y JavaScript hace el resto que es extremadamente significativo.

Más allá de esta integración, la estructura sigue siendo parte esencial de un documento. La misma provee los elementos necesarios para ubicar contenido estático o dinámico, y es también una plataforma básica para aplicaciones. Con la variedad de dispositivos para acceder a internet y la diversidad de interfaces disponibles para interactuar con la web, un aspecto básico como la estructura se vuelve parte vital del documento. Ahora la estructura debe proveer forma, organización y flexibilidad, y debe ser tan fuerte como los fundamentos de un edificio (Gauchat, 2012).

1.3.4. Entorno integrado de desarrollo

Un Entorno de Desarrollo Integrado (Integrated Development Environment y por sus siglas en inglés IDE), es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

Los IDE ofrecen un marco de trabajo para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto (Anderson, y otros, 2014).



1.3.4.1. NetBeans 8.0

Es una herramienta utilizada por los programadores, pensada para escribir, compilar, depurar y ejecutar programas. Esta herramienta está codificada en el lenguaje de programación Java y permite desarrollar aplicaciones web, de escritorio y móviles usando las plataformas de Java. Es libre y gratuito para uso comercial como no comercial, sin restricciones para su uso. Está soportado por una activa comunidad de desarrolladores que ofrece una amplia documentación y recursos de entrenamiento. Es multiplataforma, disponible para diversos sistemas operativos como Windows, Mac OS y GNU Linux. Su instalación y actualización es muy simple y una vez instalado se le pueden adicionar módulos que permiten extender sus funcionalidades (Netbeans, 2015).

1.3.5. Herramienta de modelado

Las herramientas de modelado permiten enfocarse en las características más importantes que posee un sistema. Con el uso de estas herramientas se puede discutir todos los cambios y correcciones en los requisitos, teniendo en cuenta un costo y riesgo mínimo. Además, se puede verificar que el analista comprenda correctamente el ambiente del cliente y que haya sido respaldado con información para que los diseñadores y programadores puedan construir el sistema sin ninguna dificultad.

1.3.5.1. Visual Paradigm 8.0

Es una herramienta CASE (Ingeniería de Software Asistida por Computación). La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta de software de probada utilidad para el analista. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque orientado a objetos. En el presente trabajo se utiliza para representar el diagrama de clases, el diagrama de despliegue y el diagrama de componentes. Esta herramienta está especializada en la ingeniería del software de bases de datos. Incluye herramientas para la ingeniería inversa de bases de datos (Burrows, 2010).



1.3.6. Sistema gestor de bases de datos

Un sistema gestor de base de datos (SGBD), es una agrupación de programas que sirven para definir, construir y manipular una base de datos. En la manipulación de una base de datos, los SGBD deben incluir un control de concurrencia, o sea, deben permitir a varios usuarios tener acceso "simultáneo" a la base de datos. Controlar la concurrencia implica que, si varios usuarios acceden a la base de datos, la actualización de los datos se haga de forma controlada para que no haya problemas. Un SGBD también debe encargarse de cumplir las reglas de integridad y redundancias. Otra función importante en un SGBD es su capacidad de realizar copias de seguridad y de recuperación de datos. Además, permiten la restricción de accesos no autorizados al suministrar múltiples interfaces de usuario (Alegsa, 2010).

1.3.6.1. MySQL 5

Utiliza el lenguaje de consulta estructurado (por sus siglas en inglés SQL). Es un producto de código abierto desarrollado bajo la licencia GPL (Licencia Pública General GNU), lo que hace posible que pueda ser modificado por los usuarios que lo utilizan. Es un SGBD usado y estandarizado para acceder a base de datos relacionales. Facilita la integración con programas desarrollados en C y C++ pues fue desarrollado en este lenguaje. Es un sistema multiplataforma, totalmente funcional en sistemas operativos como Linux, Mac X, UNIX y Microsoft Windows. Permite la creación de base de datos con acceso desde páginas web dinámicas, dando la posibilidad de realizar múltiples y rápidas consultas (Thibaud, 2006).

Para el desarrollo de la propuesta de solución se utiliza este sistema gestor de bases de datos, pero una vez lista, esta podrá ser instalada en cualquier otro sistema gestor de bases de datos, puesto que técnicamente, las consultas que se realizan se hacen utilizando la capa de abstracción de Drupal, por lo que no influye en el correcto funcionamiento del módulo, el sistema gestor de bases de datos que se utilice, una vez este implementada esta.

1.3.7. Servidor web

La principal tarea de un servidor web es alojar sitios y/o aplicaciones, las cuales son accedidas por los clientes utilizando un navegador que se comunica con el servidor mediante el protocolo HTTP. El servidor espera peticiones del cliente y le da respuesta con el contenido solicitado. Disponen de intérpretes de lenguajes de programación que ejecutan códigos embebidos dentro del HTML de las páginas que contiene el sitio antes de enviar el resultado al cliente (Arredondo Morales, 2009).

1.3.7.1. Apache 2.4

Apache es un servidor web utilizado para el protocolo HTTP, posee la capacidad de funcionar como un proceso independiente, sin necesitar el apoyo de otras aplicaciones o directamente del usuario. Es una herramienta de software libre y código abierto desarrollado bajo una variante de la licencia BSD⁵, modular, multiplataforma, extensible, popular (cuenta con un gran soporte y ayuda técnica en la web) y gratuito. El servidor web Apache es altamente configurable y posee gestión de logs. Permite además la personalización de respuestas ante errores que puedan ocurrir (Martínez Guaita, 2012).

Dada su fácil integración con el CMS Drupal, además es una herramienta de software libre y código abierto, modular, multiplataforma, extensible, con amplio soporte y ayuda técnica, gratuito y por todas las características antes expuestas que lo destacan como el idóneo se decide escoger el servidor web Apache en su versión 2.4.

1.3.8. Herramientas de pruebas

Una excelente forma de validar las aplicaciones y sitios web consiste en someterlos a pruebas para comprobar cómo se comportará su rendimiento general y el cumplimiento respecto a las especificaciones iniciales del sistema. Una prueba tiene éxito si descubre un error no detectado hasta entonces.

1.3.8.1. Escáner de vulnerabilidades web Acunetix 8.0

El escáner de vulnerabilidades web Acunetix es una herramienta que será capaz de escanear sitios web en busca de posibles fallos de seguridad que puedan poner en peligro la integridad de la página publicada en internet. Esta aplicación ejecuta una serie de pruebas, totalmente configurables por el usuario, para identificar las vulnerabilidades tanto en la programación de la página como en la configuración del servidor. Detecta técnicas de hacking como pueden ser: ataques de ejecución de código, inyección SQL, autenticación, entre otros. Además, destaca por su completo conjunto de herramientas: editor HTTP, Sniffer HTTP, Flooder HTTP, encriptador de palabras, control de vulnerabilidades, SQL, entre otros (Acunetix, 2016).

⁵ (Berkeley Software Distribution o en su concepto en español distribución de software Berkeley).



1.3.8.2. Apache JMeter 2.12

Apache JMeter es una de las herramientas Open Source para realizar pruebas de carga. Además de HTTP soporta protocolos como POP3, IMAP, FTP, entre otros y permite programar lo que llama planes de prueba que consisten en guiones de ejecución para las pruebas de stress. Es un software de código abierto, una aplicación diseñada para cargar la conducta funcional de prueba y medida de rendimiento. Fue diseñado originalmente para aplicaciones web de prueba, pero desde entonces se ha expandido a otras funciones de prueba. Jmeter puede ser utilizado para probar el rendimiento tanto en los recursos estáticos y dinámicos. Se puede utilizar para simular una carga pesada en un servidor, grupo de servidores, la red o el objeto a probar su resistencia o para analizar el rendimiento general bajo diferentes tipos de carga. Se puede utilizar para realizar un análisis gráfico de rendimiento o para probar su comportamiento (Apache Software Foundation, 2016).

1.4. Conclusiones del capítulo

- El estudio de los sistemas homólogos existentes relacionados con la presente investigación evidenció que las limitantes encontradas en los diferentes sistemas estudiados, tanto a nivel nacional como internacional, para la gestión logística de eventos académicos y científicos, conllevaron a la necesidad de desarrollar una nueva propuesta de solución.
- El estudio realizado de las principales herramientas, tecnologías y lenguajes de programación a utilizar, permitió realizar la selección de los más adecuados para dar solución al problema planteado y se seleccionó Drupal 7, como sistema gestor de contenido, NetBeans 8.0 como entorno de desarrollo integrado, como servidor web Apache y como sistema gestor de bases de datos MySQL 5. El proceso de desarrollo estuvo guiado por la metodología AUP-UCI.



Capítulo II: Características y arquitectura del módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI

Introducción

En este capítulo se realiza una descripción general de la propuesta de solución, a partir de su funcionamiento. Se representan a través de un modelo conceptual los principales conceptos que se manejan actualmente durante los procesos de gestión logística de eventos académicos y científicos en la UCI. Se define la arquitectura y los patrones arquitectónicos aplicables al sistema de gestión de contenidos seleccionado. Se definen los requisitos funcionales y no funcionales para el módulo a desarrollar. Las funcionalidades se describen mediante las historias de usuario (HU).

2.1. Descripción de la propuesta de solución

El módulo de gestión logística de eventos para el Sistema de Gestión Multieventos, permitirá la gestión de eventos de la UCI. Utilizando este módulo se podrá crear, modificar o eliminar la información asociada a dichos eventos. El módulo permitirá realizar solicitudes de alimentación, hospedaje, locales, materiales y de transporte. También, brindará la opción de exportar a pdf las solicitudes creadas, para que, posteriormente, sean enviadas por correo electrónico, al ente responsable de procesar cada una de ellas, para la realización y el cumplimiento de dichos eventos.

Para la gestión logística de eventos académicos y científicos, se define el rol “Administrador”:

Rol Administrador: Puede acceder a todas las funcionalidades del sistema (Dirección de Ciencia, Tecnología e Innovación).

2.2. Requisitos de software

El objetivo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto (Rodríguez, 2015). Dentro de las tareas de esta disciplina está la Especificación de Requisitos de Software (ERS) que constituye un medio de comunicación entre clientes, usuarios, ingenieros de requisitos y desarrolladores. En la ERS deben recogerse tanto las necesidades de clientes y usuarios (necesidades del negocio, también conocidas como requisitos de usuario, requisitos de cliente y necesidades de usuario) como los requisitos que debe cumplir el software a desarrollar para satisfacer todas las necesidades (requisitos del producto, también conocidos como requisitos de sistema o requisitos de software) (Creed,

2017). A continuación, se listan los requisitos funcionales y no funcionales definidos para la implementación de la propuesta de solución como resultado de entrevistas realizadas al cliente.

2.2.1. Requisitos Funcionales

Un requisito funcional es la capacidad que debe tener un software para que sea útil en la realización de los procesos de negocio de una organización (Creed, 2017).

A continuación, en la **tabla 2** se listan los requisitos funcionales de la propuesta de solución, la prioridad para el cliente y su complejidad.

Tabla 2: Requisitos Funcionales (Elaboración Propia).

| Requisitos funcionales | Nombre del requisito funcional | Prioridad | Complejidad |
|------------------------|---|-----------|-------------|
| RF1 | Crear solicitud de alimentación | Alta | Baja |
| RF2 | Editar solicitud de alimentación | Alta | Baja |
| RF3 | Listar solicitud de alimentación | Baja | Baja |
| RF4 | Visualizar solicitud de alimentación | Alta | Media |
| RF5 | Eliminar solicitud de alimentación | Media | Baja |
| RF6 | Exportar solicitud de alimentación a pdf | Baja | Alta |
| RF7 | Enviar solicitud de alimentación por correo electrónico | Alta | Alta |
| RF8 | Crear solicitud de hospedaje | Alta | Baja |
| RF9 | Editar solicitud de hospedaje | Alta | Baja |
| RF10 | Listar solicitud de hospedaje | Baja | Baja |
| RF11 | Visualizar solicitud de hospedaje | Alta | Media |
| RF12 | Eliminar solicitud de hospedaje | Media | Baja |

*Módulo de gestión logística para el Sistema de Gestión
Multieventos de la UCI.*



| | | | |
|-------------|---|-------|-------|
| RF13 | Exportar solicitud de hospedaje a pdf | Baja | Alta |
| RF14 | Enviar solicitud de hospedaje por correo electrónico | Alta | Alta |
| RF15 | Crear solicitud de locales | Alta | Baja |
| RF16 | Editar solicitud de locales | Alta | Baja |
| RF17 | Listar solicitud de locales | Baja | Baja |
| RF18 | Visualizar solicitud de locales | Alta | Media |
| RF19 | Eliminar solicitud de locales | Media | Baja |
| RF20 | Exportar solicitud de locales a pdf | Baja | Alta |
| RF21 | Enviar solicitud de locales por correo electrónico | Alta | Alta |
| RF22 | Crear solicitud de materiales | Alta | Baja |
| RF23 | Editar solicitud de materiales | Alta | Baja |
| RF24 | Listar solicitud de materiales | Baja | Baja |
| RF25 | Visualizar solicitud de materiales | Alta | Media |
| RF26 | Eliminar solicitud de materiales | Media | Baja |
| RF27 | Exportar solicitud de materiales a pdf | Baja | Alta |
| RF28 | Enviar solicitud de materiales por correo electrónico | Alta | Alta |
| RF29 | Crear solicitud de transporte | Alta | Baja |
| RF30 | Editar solicitud de transporte | Alta | Baja |
| RF31 | Listar solicitud de transporte | Baja | Baja |



| | | | |
|-------------|---|-------|-------|
| RF32 | Visualizar solicitud de transporte | Alta | Media |
| RF33 | Eliminar solicitud de transporte | Media | Baja |
| RF34 | Exportar solicitud de transporte a pdf | Baja | Alta |
| RF35 | Enviar solicitud de transporte por correo electrónico | Alta | Alta |

A continuación, se muestra en la **tabla 3** el resumen de la etapa de captura de los requisitos funcionales.

Tabla 3. Resumen de la etapa de captura de los RF (Elaboración Propia).

| Prioridad para el cliente | Cantidad de requisitos funcionales |
|----------------------------------|---|
| Alta | 20 |
| Media | 5 |
| Baja | 10 |
| Total | 35 |

2.2.2. Requisitos no funcionales

Los requisitos no funcionales son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de recursos y del proceso de desarrollo, como impuestas por los estándares. Los requisitos no funcionales se aplican al sistema como un todo, más que a características o a servicios individuales del mismo (Creed, 2017).

Se obtuvo un total de diecisiete (17) requisitos no funcionales, distribuidos en especificaciones de usabilidad, seguridad, soporte, interfaz de hardware, interfaz de software, requisitos legales, operacionales y rendimiento, los cuales se relacionan a continuación:

Usabilidad

1. El módulo será fácil de usar al ser visualizado en dispositivos para las resoluciones 320x480, 768x1024, 1024x980 y 1325x980.



2. Se utilizarán tamaños de letra apropiados para personas con alguna discapacidad visual. Se hará uso de la tecla Control del teclado y el scroll del mouse para la lectura de pantallas por personas con discapacidad visual.

Seguridad

3. El sistema será utilizado solo por los usuarios con permisos de usuario administrador y usuario registrado con permisos.
4. Los errores deben mostrar la menor cantidad de detalles posible, para evitar brindar información que comprometa la seguridad e integridad del sistema.

Soporte

5. El sistema funcionará en varios sistemas operativos como Windows XP,7,8,8.1,10, Linux en todas las distribuciones basadas en Debian, Ubuntu, OpenSuse, Fedora. Interfaz de Hardware.

Interfaz de Hardware

6. PC cliente con una capacidad de memoria RAM: 2 GB.
7. PC Servidor web y Servidor de BD CPU: 2 núcleos de procesamiento. Al menos 4 GB de memoria. RAM Espacio: Mínimo 20 GB.

Interfaz de Software

8. El menú de configuración del módulo presentará una interfaz agradable, utilizando colores claros, en tonalidad azul.
9. Los textos serán presentados en tipografías compatibles con varios sistemas operativos y se resaltarán los encabezados en negrita.

Requisitos Legales

10. Las tecnologías y herramientas utilizadas están bajo licencia GNU/GPL en su versión 2 para el CMS Drupal.
11. Uso de licencia PHP License.
12. Uso de licencia de Visual Paradigm en su versión 8.0.

Operacionales

13. El sistema necesitará para su correcto funcionamiento cargar las librerías fpdf, tcpdf.
14. Para la distribución del producto y su instalación se necesitará un entorno de trabajo compuesto por:
 - Servidor web: Apache, Nginx, o Microsoft IIS.
 - Base de Datos: MySQL v.5.0, PostgreSQL v.9.3, MariaDB.



- Sistema Gestor de contenido Drupal v.7.54.
- Lenguaje de programación PHP v.5.2.5 o versiones posteriores recomendadas.
- Navegadores web Chrome v.48, Firefox v.59, Opera v.34 e Internet Explorer a partir de su versión 10.

Rendimiento

15. El sistema debe responder en un máximo de 5 segundos las solicitudes de los usuarios.
16. El sistema debe soportar un mínimo de 100 peticiones concurrentes.
17. El sistema debe ser escalable permitiendo que se le pueden agregar nuevas funcionalidades sin afectar las funcionalidades implementadas.

2.3. Historias de usuario

La metodología AUP-UCI, en su escenario 4 para la disciplina requisitos, genera las Historias de Usuario (HU), estas se utilizan para especificar los requisitos de software en la aplicación. Las HU se descomponen en tareas de programación y describen las características que el sistema debe cumplir, están escritas en un formato legible por el cliente, sin necesidad de sintaxis técnicas (Rodríguez, 2015).

Cada historia de usuario debe ser lo suficientemente comprensible y delimitada para que se pueda implementar en unas tres semanas aproximadamente (para no superar el tamaño de una iteración, que es el tiempo estimado para una entrega de un componente de desarrollo de manera incremental (Villamizar Suaza, 2013).

Estas quedan estructuradas de la siguiente forma:

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: Grado de prioridad que le asigna el cliente a la HU en dependencia del valor en el negocio. Los valores que puede tomar son (Alta, Media o Baja).

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

Programador: Nombre del programador encargado de desarrollar la HU.

Tiempo estimado: Esfuerzo estimado por el equipo de desarrollo para darle cumplimiento a la HU.

Tiempo real: Plazo real que el equipo de desarrollo tiene para dar cumplimiento a la HU.

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

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

A continuación, se muestra en la **tabla 4** y **tabla 5** las historias de usuario utilizadas en la propuesta de solución para los requisitos Añadir medio AFT y Editar medio útil respectivamente.

Tabla 4. HU_ Eliminar solicitud de locales (Elaboración Propia).

| Historia de Usuario | |
|--|--|
| Número: 20 | Nombre: Eliminar solicitud de locales |
| Prioridad: Alta | Iteración Asignada: 2 |
| Programador: Roynel Mena Chávez | |
| Tiempo estimado: 10 horas | Tiempo real: 10 horas |
| Riesgo en desarrollo: Alto | |
| Descripción: <ul style="list-style-type: none"> El módulo deberá permitir al usuario con rol administrador eliminar las solicitudes de locales. | |
| Observaciones: <ol style="list-style-type: none"> 1. Para eliminar una solicitud, el módulo permitirá a los usuarios con roles administrador, eliminar las solicitudes en el módulo. 2. El módulo mostrará una ventana con el siguiente mensaje: ¿Está seguro de que quiere eliminar Solicitud de Locales? 3. Luego de que se haya eliminado la solicitud, el sistema mostrará el siguiente mensaje: “La acción se ha realizado satisfactoriamente”. | |

Módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI.



Prototipo de interface:

Añadir contenido **Find content** Tipos de contenido Cron Rendimiento Editar atajos

[+ Añadir contenido](#)

MOSTRAR SÓLO LOS ELEMENTOS EN LOS QUE

estado

tipo

idioma

ACTUALIZAR OPCIONES

Publicar el contenido seleccionado

| <input type="checkbox"/> | TÍTULO | TIPO | AUTOR | ESTADO | ACTUALIZADO | IDIOMA | OPERACIONES |
|--------------------------|---------------------------|-------------------------------------|----------|-----------|----------------------|---------|------------------------|
| <input type="checkbox"/> | Solicitud de Hospedaje | Solicitud de hospedaje | rmchavez | publicado | 13/11/2019 - 2:55pm | Español | editar eliminar |
| <input type="checkbox"/> | Solicitud de Transporte | Solicitud de transporte | rmchavez | publicado | 11/11/2019 - 11:24am | Español | editar eliminar |
| <input type="checkbox"/> | Solicitud de Locales | Solicitud de locales | rmchavez | publicado | 06/11/2019 - 11:42pm | Español | editar eliminar |
| <input type="checkbox"/> | Solicitud de Alimentación | Solicitud de servicios de protocolo | rmchavez | publicado | 06/11/2019 - 2:10pm | Español | editar eliminar |
| <input type="checkbox"/> | Solicitud de Materiales | Solicitud de materiales | rmchavez | publicado | 05/11/2019 - 1:21am | Español | editar eliminar |

Panel de control Contenido Estructura Apariencia Usuarios Módulos Configuración Informes Ayuda

Añadir contenido Find content Tipos de contenido Cron Rendimiento

Inicio » Solicitud de Locales

¿Está seguro de que quiere eliminar *Solicitud de Locales*?

Esta acción no se puede deshacer.

Tabla 5. HU_ Crear solicitud de transporte (Elaboración Propia).

| Historia de Usuario | |
|---|--|
| Número: 30 | Nombre: Crear solicitud de transporte |
| Prioridad: Alta | Iteración Asignada: 1 |
| Programador: Roynel Mena Chávez | |
| Tiempo estimado: 1.15 h | Tiempo real: 1.15 h |
| Riesgo en desarrollo: --- | |
| Descripción: <ul style="list-style-type: none"> El módulo deberá permitir al usuario con rol administrador crear la solicitud de transporte. | |
| Observaciones: <p>Para crear una solicitud el módulo permitirá a los usuarios con roles administrador, añadir contenidos en el módulo. Para ello se deben introducir los siguientes datos:</p> <ul style="list-style-type: none"> Dependencia (obligatorio, referencia a término, lista de selección) Ejemplo: Centro de Ideoinformatica. Fecha de solicitud (obligatorio, campo de tipo fecha). Solicitud (obligatorio, fieldcollection). Solicitado por (obligatorio, campo de tipo texto). Aprobado por (obligatorio, campo de tipo texto). <p>Luego de que se haya insertado el evento el módulo mostrará el siguiente mensaje: “Solicitud de transporte se ha creado”.</p> | |



Prototipo de interface:

Añadir contenido Find content Tipos de contenido Cron Rendimiento Editor atajos

DEPENDENCIA *
- Seleccione un valor -
En este campo usted debe seleccionar el área que solicita la Solicitud de Transporte.

FECHA DE LA SOLICITUD *
11/26/2019
E.g., 11/27/2019

SOLICITUD * [Mostrar pesos de la fila](#)

| NO. * |
|----------------------|
| <input type="text"/> |
| <input type="text"/> |

[Añadir otro elemento](#) [Mostrar pesos de la fila](#)

FECHA *

| FECHA * |
|----------------------|
| <input type="text"/> |

2.4. Arquitectura

El secreto de Drupal para conseguir su reconocida flexibilidad y facilidad en la creación de sitios web es la abstracción y organización en capas que aplica en el tratamiento de los contenidos. En lugar de considerar el sitio web como un conjunto de páginas interrelacionadas, Drupal estructura los contenidos en una serie de elementos básicos. Estos son los nodos (nodes), módulos (modules), bloques y menús (blocks & menus), permisos de usuario y plantillas (templates) (Borrueal, 2014).

2.4.1. N-Capas de Drupal

Al asumirse Drupal 7 como CMS para el desarrollo de la presente investigación se hereda la arquitectura n-capas de Drupal 7. Lo que permite explotar las potencialidades que brinda dicha arquitectura, donde el objetivo principal es poder intercambiar porciones de la aplicación sin necesidad de modificarla completamente, esto además permite ampliar funcionalidades e integración de nuevos módulos y separar los diferentes aspectos del desarrollo mediante capas, divididas específicamente en 5, que son descritas a continuación:

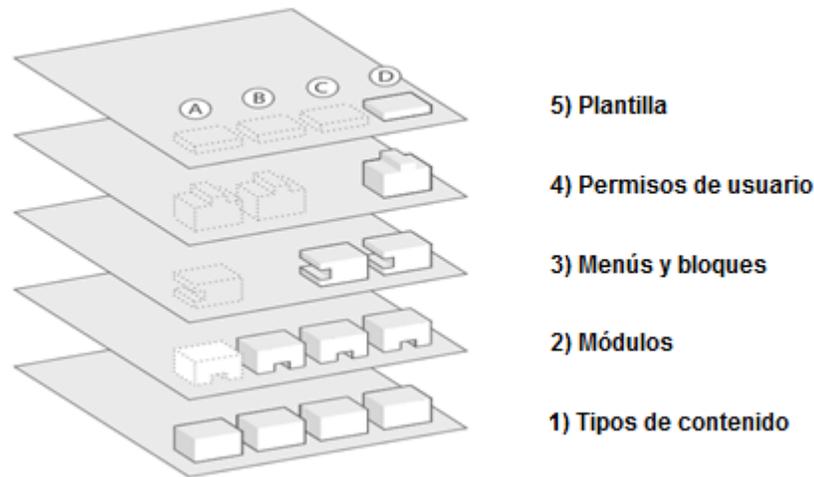


Figura 2. Arquitectura de la aplicación (Drupal. 2016).

Una primera capa (Tipos de contenido), para los nodos, donde se ubican los tipos de contenidos (solicitud de alimentación, solicitud de hospedaje, solicitud de locales, solicitud de materiales y solicitud de transporte), en esta capa se utiliza un conjunto de funciones llamadas hook o ganchos como también se les conoce, estos son el medio de comunicación entre los módulos y el sistema. En la segunda capa (Módulos), es donde se ubican los módulos, en este caso, el módulo que conforma la propuesta de solución, el cual lleva por nombre módulo gestión, y además, otros módulos que fueron necesarios para el desarrollo de la investigación. En la tercera capa (Menús y bloques), se implementa el hook menú que permite gestionar los menús del sistema, para crear los enlaces a la página de los contenidos de gestión logística. Una cuarta capa (Permisos de usuario), para los permisos de usuarios, donde se creó el permiso “acceso a gestión logística”, que puede ser asignado al usuario encargado de gestionar las solicitudes, y la quinta capa (Plantillas), dedicada a darle formato a la interfaz de usuario, con estilos CSS, mediante los archivos tpl.

2.5. Patrones de diseño

Los patrones de diseño representan la descripción de un problema particular y recurrente, que aparece en contextos específicos y presenta un esquema genérico demostrado con éxito para su solución; este último se especifica mediante la descripción de los componentes que la constituyen, sus responsabilidades y desarrollos, así como también la forma como estos colaboran entre sí. Constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son descripciones de clases

cuyas instancias colaboran entre sí. La clave para la reutilización es anticiparse a los nuevos requisitos y cambios, de modo que los sistemas evolucionen de forma adecuada. Facilitan la reusabilidad, extensibilidad y mantenimiento (Quimera, 2015).

2.5.1. Patrones GoF

Los patrones GoF⁶, describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad. Permiten crear grupos de objetos para ayudar a realizar tareas complejas (Márquez, 2013).

Los patrones usados para el desarrollo del sistema se evidencian de la forma siguiente:

Observer (Observador): Los módulos que implementan un hook de inserción o actualización de una entidad, son declarados como observadores de las entidades con las que interactúan (Silvente Lao, 2015). En la aplicación el uso del patrón Observador se evidencia cuando se modifica el contenido solicitud en el módulo, todas las dependencias del sistema son informadas de la modificación realizada.

Bridge (Puente): La capa de abstracción de datos de Drupal se encuentra implementada siguiendo el patrón puente. El módulo implementado es programado de manera que sea independiente del motor de base de datos que utiliza el sistema. Esto se logra por la capa de abstracción de base de datos, sobre la que se pueden desarrollar consultas siguiendo la API definida (Silvente Lao, 2015). En la aplicación el uso del patrón puente se evidencia al eliminar un contenido, pues impide que se afecten los módulos relacionados con el contenido eliminado, permitiéndoles a estos módulos seguir funcionando y que puedan seguir siendo utilizados por otros contenidos.

Chain of responsibility (Cadena de responsabilidades): El sistema de menú de Drupal es la evidencia del patrón cadena de responsabilidades. En cada petición de una página, el sistema de menú de Drupal determina si hay algún módulo para responder la petición, si el usuario tiene acceso al recurso solicitado y que función se debe llamar para procesar la petición. En este proceso se transmite el mensaje de la petición por cada uno de los componentes que se encuentran inmersos (Silvente Lao, 2015).

⁶Banda de los cuatro, compuesta por los siguientes autores: Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides.



2.6. Modelo de despliegue

Un modelo de despliegue consiste en una representación estructural de la arquitectura del sistema desde el punto de vista de la distribución de los artefactos del software en los destinos de despliegue; definiendo a los artefactos como representaciones de elementos concretos en el mundo físico que son el resultado de un proceso de desarrollo (Sarmiento, 2016).

A continuación, en la **figura 3** se muestra el modelo de despliegue que resulta de la presente investigación.

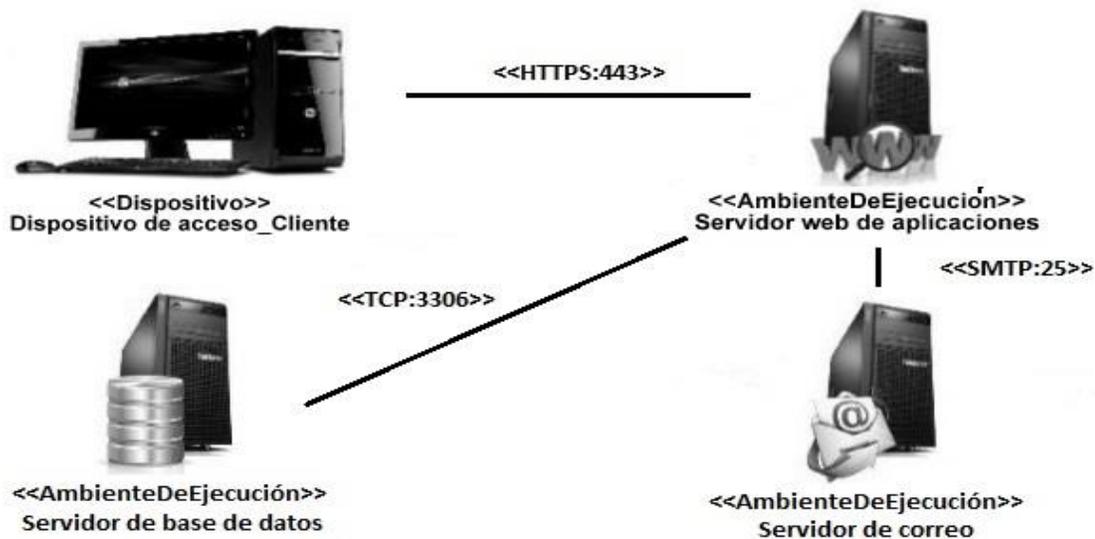


Figura 3. Modelo de Despliegue (Elaboración Propia)

2.6.1. Descripción de los elementos del modelo de despliegue

Dispositivo de acceso_Cliente: La estación de trabajo necesita un navegador web para conectarse al sistema hospedado en el servidor de aplicaciones utilizando el protocolo de comunicación HTTP/HTTPS.

Servidor de aplicaciones: Es la estación de trabajo que hospeda el código fuente de la aplicación y que le brinda al usuario las interfaces para realizar los procesos del sistema. Esta estación se comunica con el



servidor de base de datos donde se almacenan los datos de la aplicación realizando la comunicación mediante el protocolo TCP/IP.

Servidor de BD: Este servidor es el encargado del almacenamiento de los datos del sistema. Se comunica con el servidor de aplicaciones del sistema, posibilitando el acceso mediante el usuario con privilegios para las operaciones determinadas a realizarse en el mismo.

Servidor de correo: Este servidor es el encargado del envío de correos electrónicos comunicándose a través del protocolo SMTP.

HTTPS: Protocolo para establecer a través del puerto 443 la conexión segura entre el dispositivo de acceso cliente y el servidor de aplicaciones. La conexión es por cable vía modem, LAN o red inalámbrica con una velocidad de más de 64 Kbps.

TCP: Protocolo para establecer la conexión entre el servidor de aplicaciones y el servidor de base de datos. Para el servidor de base de datos de *MySQL* se define el puerto 3306. La conexión entre el servidor web y el servidor de base de datos permite dar órdenes y obtener información de esta.

SMTP: El protocolo simple de transferencia de correo es un protocolo de red utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos a través del puerto 25.

2.7. Modelo de Datos

Un modelo de datos es un conjunto de conceptos y reglas que permiten estructurar los datos resultantes de la observación de la realidad, de forma que pueden ser representadas todas sus propiedades, tanto estáticas como dinámicas. Los modelos de datos de alto nivel o conceptuales utilizan conceptos muy cercanos a la forma en que los usuarios perciben los datos, mientras que los de bajo nivel o físicos describen detalles de cómo se almacenan los datos en la computadora (Gómez, 2015).

Módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI.

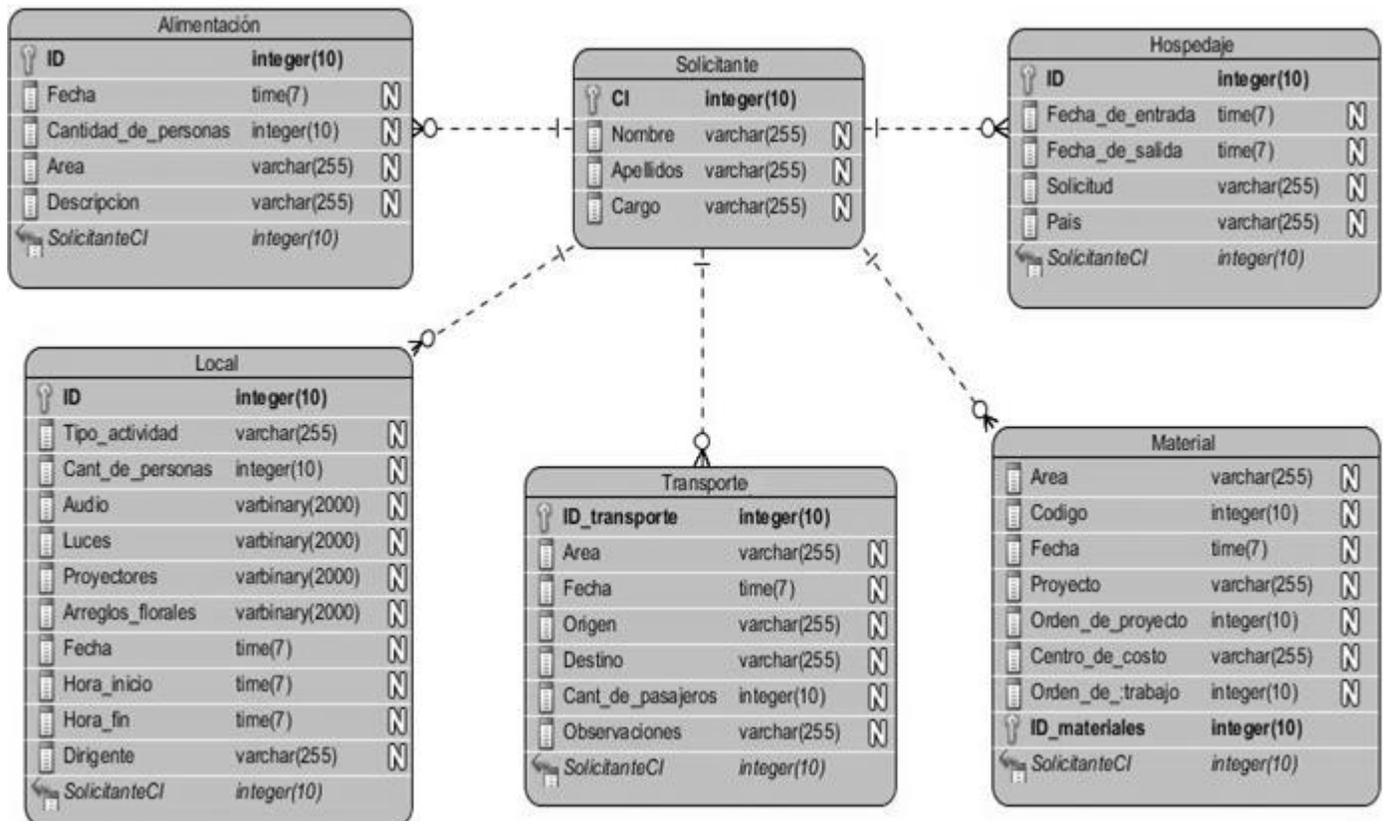


Figura 4. Modelo de datos (Elaboración propia).

2.8. Conclusiones del capítulo

- La obtención de los requisitos funcionales y no funcionales obtenidos a partir del proceso de identificación de los requisitos, garantizó que la solución respondiera a las necesidades del cliente y de los usuarios finales, sirviendo de guía para el desarrollo de las distintas funcionalidades del módulo.
- La selección de la arquitectura y los patrones de diseño utilizados por Drupal permitieron ampliar funcionalidades e integración de nuevos módulos.

Capítulo III: Implementación y validación del módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI.

Introducción

A partir de los artefactos generados en el capítulo anterior, se procede a la construcción del sistema. Se elabora el diagrama de componentes el cual representa cómo un sistema de software es dividido en componentes y las dependencias entre ellos. Para asegurar una mayor precisión y objetividad de la solución propuesta, se plasman los resultados obtenidos al aplicar una estrategia de pruebas con el fin de descubrir y corregir errores.

3.1. Diagrama de componentes

El diagrama de componentes permite visualizar la estructura de un sistema informático atendiendo a las partes que lo conforman. Cada componente debe ser tratado como una unidad de composición independiente e indispensable dentro de un sistema, pudiéndose encontrar dependencias entre estos. Algunos ejemplos de componentes físicos lo constituyen los archivos, módulos, librerías, ejecutables y binarios (Sommerville, 2011). Este diagrama provee una vista arquitectónica del sistema el cual permite lograr una mejor visualización de la implementación. Su realización posibilita tomar decisiones respecto a las tareas de implementación y los requisitos. A continuación, se describen los componentes del diagrama:

- **index.php:** Constituye el punto de inicio de la aplicación, es a partir de esta entrada que son solicitados los distintos módulos presentes en el CMS Drupal 7.
- **sites:** En este directorio se incorporan los temas, módulos y librerías que se puedan añadir.
- **default:** Contiene las clases de configuración settings.php.
- **scripts:** Contiene archivos .sh que extienden las funcionalidades de Drupal 7.
- **misc:** Contiene imágenes requeridas por el sistema y archivos JavaScript.
- **themes:** Contiene los temas correspondientes a la distribución de Drupal.
- **modules:** Contiene los módulos del núcleo.

Módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI.



- **includes:** Las funciones y scripts posibilitan el funcionamiento de los módulos y del CMS Drupal, contiene además funciones que permite la conexión a la base datos.
- **db:** Es la base datos de Drupal. Gestiona el acceso a la información almacenada referente al funcionamiento del sistema.
- **module:** Contiene la Propuesta de solución.
- **modulo_gestion.info:** Contiene información sobre el módulo (nombre, descripción, versión de Drupal, entre otros).
- **modulo_gestion.install:** Archivo que implementa el hook_schema para la creación de los tipos de contenidos: revista, evento y misión en la base de datos del sistema.
- **modulo_gestion.module:** Archivo que contiene las llamadas a las funciones hook de Drupal y algunas funciones básicas del sistema.

A continuación, en la **figura 5** se muestra el diagrama de componentes.

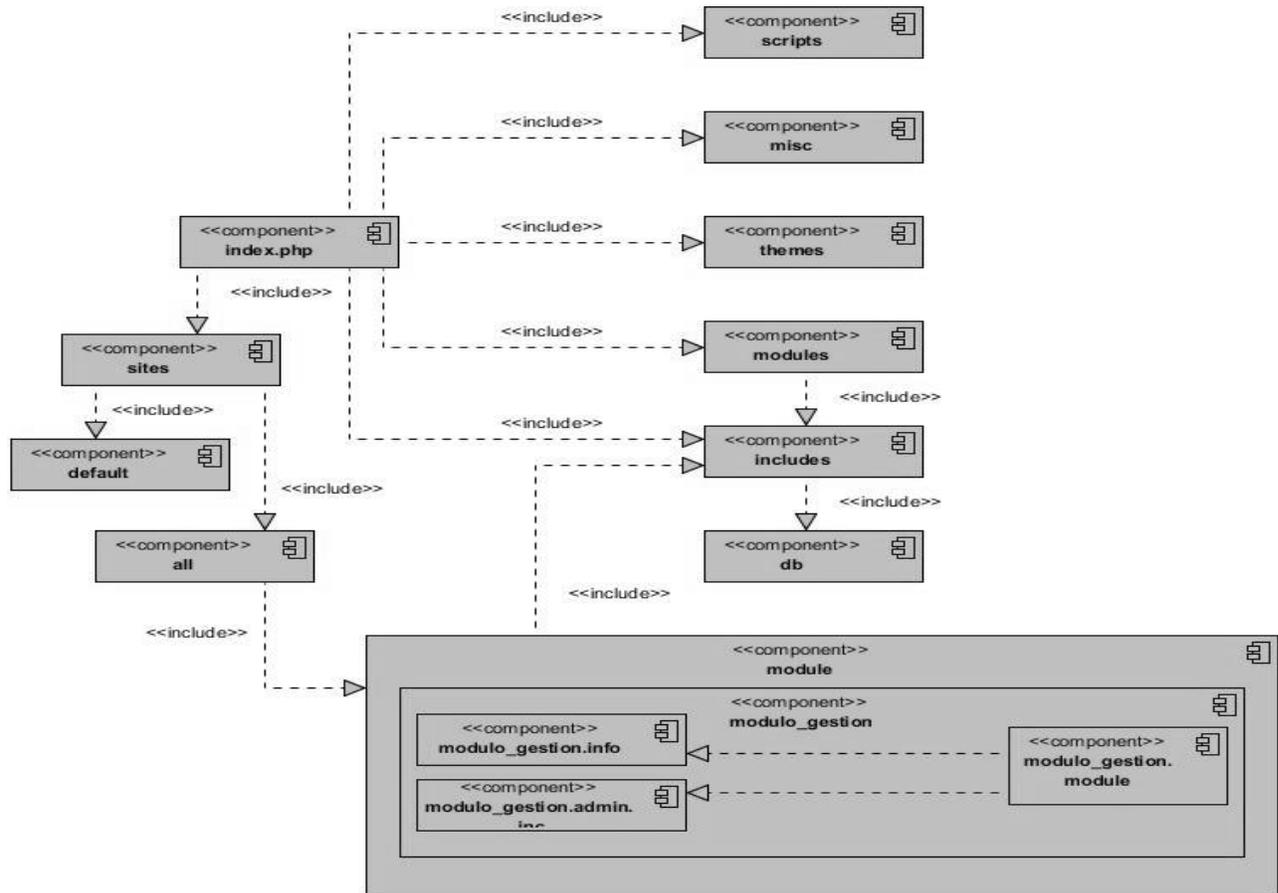


Figura 5. Diagrama de componentes (Elaboración propia).

3.2. Estándares de codificación

Los estándares de codificación o estilos de programación, como también se les llama, definen la estructura y apariencia física del código, lo que facilita la lectura, comprensión y mantenimiento del mismo (Drupal, 2016). Para el desarrollo del módulo, se utilizaron los estándares de codificación que se describen a continuación:

3.2.1. Indentación y espacios en blanco

Para facilitar la comprensión del código se utilizan tabuladores o espacios en blanco en determinadas líneas del mismo. Las líneas no deben tener espacios en blanco al final. Se utiliza una indentación de dos espacios como se muestra en la **figura 6**.

```
174 ▼ <table class="borderless">
175 ▼   <tr>
176     <td><?php print render($content['field_nombre_y_apellidos_l']); ?></td>
177 ▼   <td><b>Firma:</b>
178     <div class="firma"/>
179   </td>
180   <td><?php print render($content['field_fecha_l']); ?></td>
181 </tr>
182 </table>
```

Figura 6. Uso de indentación y espacios en blanco (Elaboración propia).

3.2.2. Operadores

Todos los operadores binarios como: +, -, =, ==, >, entre otros, deben tener un espacio antes y después del operador, para facilitar la lectura. Los operadores unarios como: ++, no deberían tener un espacio entre el operador y la variable o el número que están operando, como se muestra en la **figura 7**.

```
129 <div class = "espacio-entre-parrafos"> <?php print
render($content['field_algo_mas_']); ?></div>
```

Figura 7. Uso de operadores (Elaboración propia).

3.2.3. Estructuras de control

Las estructuras de control incluyen if, for, while, switch, entre otros. Las mismas deben tener un espacio entre la palabra clave de control y el paréntesis de apertura, para distinguirlas de las llamadas a funciones como se muestra en la **figura 8**.

```
164 ▼ foreach ($content['field_solicitud_de_hospedaje']['#items'] as $entity_uri) {
165   print "<tr>";
166   $fila = entity_load('field_collection_item', $entity_uri);
```

Figura 8. Uso de estructuras de control (Elaboración propia).

3.2.4. Longitud de líneas

Las líneas de código no deben contener más de 80 caracteres como se muestra en la **figura 9**.



```
106 <?php
107 print render($content['field_depencencia']);
108 print render($content['field_fecha_de_la_solicitud']);
```

Figura 9. Longitud de línea con menos de 80 caracteres (Elaboración propia).

3.2.5. Arreglos

Los valores dentro de un arreglo deben separarse por un espacio (después de la coma que los separa). El operador => se debe separar por un espacio a ambos lados. En caso de que la línea de declaración del arreglo supere los 80 caracteres, cada elemento se debe escribir en una única línea, indentándolo una vez (2 espacios). En este último caso, la coma de separación del último elemento también se deberá escribir, aunque no existan más elementos como se muestra en la **figura 10**.

```
169 ▼ foreach ($fila as $fila_object) {
170     print "<td>";
171     print render($fila_object->field_no_h['und'][0]['value']);
172     print "</td>";
173     print "<td>";
174     print render($fila_object->field_nombre_y_apellidos['und'][0]['value']);
175     print "</td>";
176     print "<td>";
177     print render($fila_object->field_pa_s['und'][0]['safe']);
178     print "</td>";
179     print "<td>";
180     print render($fila_object->field_pasaporte['und'][0]['value']);
181     print "</td>";
182 }
```

Figura 10. Uso de arreglos (Elaboración propia).

3.2.6. Uso de comillas

Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres. Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto o cuando el texto puede incluir alguna comilla simple, como se muestra en la **figura 11**.

```
134 <div class = "margin-right"><h6><?php print
render($content['field_entrada']); ?></h6></div>
```

Figura 11. Uso de comillas (Elaboración propia).

3.2.7. Etiquetas de apertura y cierre de PHP

Se omite la etiqueta de cierre de PHP: (`?>`) al final de los archivos: `module`, `inc`, `install`, como se muestra en la **figura 12**.

```
1  <?php
2  function modulo_bloque_admin_settings() {
~  |
```

Figura 12. Uso de etiquetas PHP (Elaboración propia).

3.2.8. Uso de punto y coma (;) en código PHP

En Drupal se hace obligatorio utilizar el terminador de línea (`;`) como se muestra en la **figura 13**.

```
106  <?php
107  print render($content['field_depencencia']);
108  print render($content['field_fecha_de_la_solicitud']);
```

Figura 13. Uso de punto y coma (Elaboración propia)

3.2.9. Nombre de funciones

Los nombres de las funciones deben escribirse en minúsculas, así como las palabras separadas por un guion bajo. En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior. Debe incluirse siempre como prefijo el nombre del módulo, como se muestra en la **figura 14**.

```
76 ▼ function modulo_bloque_info(){
```

Figura 14. Nombre de funciones (Elaboración propia).

3.2.10. Nombre de variables

Las variables se deben nombrar en minúsculas y las palabras deben estar separadas por un guion bajo, como muestra la **figura 15**.

```
80  $array_cargo = array("Elige un cargo",
```

Figura 15. Nombre de variables (Elaboración propia).

3.3. Validación de la propuesta de solución

Las pruebas son el conjunto de técnicas que permiten determinar la calidad de un producto de software. Se integran dentro de las diferentes fases del ciclo de vida del software dentro de la ingeniería de software. De esta manera, se ejecuta un programa con el objetivo de descubrir los errores que presenta. La calidad de un sistema de software es algo subjetivo que depende del contexto y del objeto que se pretenda conseguir.

Para determinar dicho nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema. Una prueba tiene éxito si descubre un error no detectado hasta entonces (García Sánchez, 2010).

3.3.1. Pruebas funcionales

Se denominan pruebas funcionales o Functional Testing, a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados. Una prueba funcional es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software (Baydes Gonzáles, 2012).

Para la ejecución de este tipo de pruebas, suelen emplearse dos métodos fundamentales: el método de Caja Blanca y el método de Caja Negra. El primero se centra en las pruebas al código de las aplicaciones; mientras que el segundo permite a los probadores enfocar su atención en el funcionamiento de la interfaz, a través del análisis de los datos de entrada y los de salida (Aleman Jimenez, y otros, 2014).

A continuación, en la **tabla 6** se describe el caso de prueba insertar evento:

Tabla 6. Caso de prueba 1 (Elaboración Propia).

| | |
|---|--|
| Caso de prueba Crear solicitud de transporte. | |
| Código de caso de prueba | Nombre de historia de usuario: Crear solicitud de transporte. |
| Nombre de la persona que realiza la prueba: Roynel Mena Chavez | |
| Descripción de la prueba: Prueba a la funcionalidad crear solicitud de transporte. | |



Entrada/Pasos de ejecución: La entrada consta de la introducción de los datos en los siguientes campos:

- Dependencia: Centro de Ideoinformatica.
- Fecha de la solicitud: 17/11/2019
- No.: 1
- Fecha: 24/11/2019
- Hora: 2:00 pm
- Origen: La Habana
- Destino: Las Tunas
- Pax: 30
- Observaciones: “.....”

Resultado esperado: Se crea la solicitud.

Evaluación de la prueba: Satisfactoria.

3.3.1.1. Resultados de las pruebas funcionales

Las pruebas funcionales aplicadas a la propuesta de solución se basaron en el diseño de casos de pruebas, donde se recogen los escenarios correspondientes a cada requisito funcional. En la primera iteración se detectaron 6 no conformidades, de ellas 5 de validación, 1 de redacción, y se corrigieron satisfactoriamente las 6 no conformidades encontradas. En la segunda iteración se encontraron un total de 4 no conformidades, de ellas 3 de validación y 1 de redacción, las cuales fueron mitigadas con éxito. Se realizó una tercera iteración donde no se detectaron no conformidades, siendo este el resultado final, lo que demuestra que el software funciona correctamente.

A continuación, en la **figura 16** se muestran los resultados obtenidos en cada iteración de prueba al módulo de gestión logística para el Sistema de Gestión Multieventos.

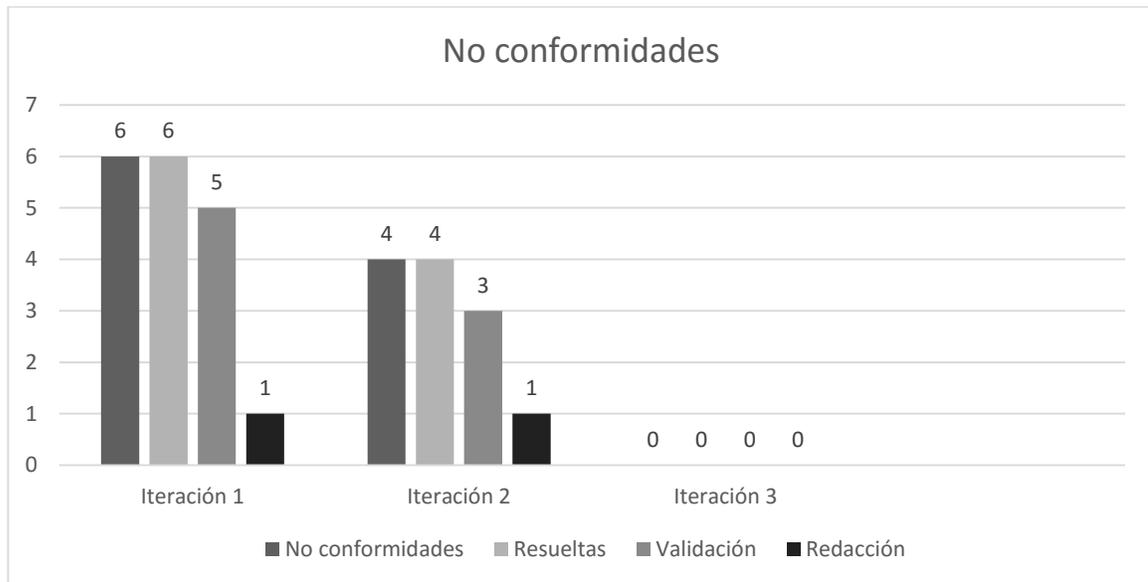


Figura 16. Comportamiento de las no conformidades por iteración (Elaboración propia).

A partir de que se le realizaron las pruebas funcionales a la propuesta de solución, se pudo verificar que las funcionalidades implementadas están en correspondencia con los requisitos funcionales y no funcionales establecidos con anterioridad. A partir de este punto de la investigación se hace necesario verificar la compatibilidad y el funcionamiento de las interfaces que comunican los componentes de la propuesta de solución, para ello se realizan las pruebas de integración.

3.3.2. Pruebas de seguridad

Las pruebas de seguridad intentan verificar que los mecanismos de protección que se construyen en un sistema lo protegerán realmente de cualquier irrupción inapropiada. Buscan medir la usabilidad, seguridad, eficiencia y portabilidad de los datos. Se diseñan para detectar las vulnerabilidades del entorno en el sistema (Pressman, 2010).

Para evaluar la seguridad de la propuesta de solución, en la presente investigación se desarrollan pruebas utilizando la herramienta Acunetix Web Vulnerability Scanner 8.0.

A continuación, en la **tabla 7** se muestra el escaneo de vulnerabilidades para pruebas de seguridad aplicadas a la propuesta de solución.

Tabla 7. Resultados de las pruebas de seguridad (Elaboración Propia).

| Categorías de vulnerabilidades | Iteración 1 | Iteración 2 |
|---|-------------|-------------|
| Formularios HTML sin protección CSRF ⁷ | 2 | 0 |
| XSS | 1 | 0 |
| Vínculos rotos | 2 | 0 |
| Total | 5 | 0 |

3.3.2.1. Resultados de las pruebas de seguridad

La prueba realizada mediante la herramienta Acunetix WVS v8 de nivel 2, arrojó que en la primera iteración se detectan 5 no conformidades, 1 alertas de riesgo medio, 2 de riesgo bajo y 2 notificaciones informativas. En la segunda iteración no se encontraron vulnerabilidades por lo que los problemas que pudieran comprometer la seguridad e integridad de la aplicación encontrados en la primera iteración fueron solucionados satisfactoriamente.

3.3.3. Pruebas de rendimiento

Las pruebas de rendimiento se implementan y se ejecutan para caracterizar y evaluar las características relacionadas con el rendimiento del destino de la prueba, como los perfiles de tiempo, el flujo de ejecución, los tiempos de respuesta y la fiabilidad. A lo largo del ciclo de vida del desarrollo de software, se implementan varios tipos de prueba de rendimiento, cada una con un objetivo de prueba diferente (Brito, 2016).

Estas pruebas se centran en identificar y eliminar los cuellos de botella de rendimiento relacionados con la arquitectura. En las iteraciones de la construcción, los tipos adicionales de pruebas de rendimiento se utilizan para ajustar el software y el entorno (optimización de los recursos y el tiempo de respuesta), y para verificar que las aplicaciones y el sistema manejan de manera aceptable las condiciones de estrés y de carga elevada, como grandes números de transacciones, clientes y/o volúmenes de datos (Brito, 2016).

Entre los diferentes tipos de pruebas de rendimiento se incluyen los siguientes (Brito, 2016):

⁷ Del inglés: **Cross-site request forgery**. Falsificación de petición en sitios cruzados.



• **Pruebas de carga:** Mediante la ejecución de las pruebas de carga es posible identificar la capacidad de recuperación de un sistema cuando es sometido a cargas variables, tanto de usuarios como de procesos. Al realizar las pruebas de carga se puede determinar el tiempo de respuesta de todas las transacciones críticas del sistema y encontrar cuellos de botella de la aplicación.

• **Prueba de estrés:** Mediante las pruebas de estrés es posible identificar la capacidad de respuesta de un sistema bajo condiciones de carga extrema, representadas por una alta concurrencia de usuarios y/o procesos. Una vez realizadas las pruebas de estrés se podrá conocer el punto de quiebre del aplicativo en términos de capacidad de respuesta, con lo cual será posible establecer acciones de optimización en diferentes niveles para asegurar una mejor capacidad de concurrencia de usuarios y procesos, que se verá reflejada en una óptima operación de negocio.

Para la realización de las pruebas de rendimiento se utilizó la herramienta JMeter, estas pruebas fueron realizadas en test de 50 hilos, los cuales simulan la cantidad de usuarios que acceden a las funcionalidades concurrentemente. Se tiene en cuenta el escenario donde se encuentra la aplicación y las condiciones de hardware y software de la computadora donde es ejecutado el muestreo; atendiendo a las siguientes características:

- Sistema Operativo: Linux Ubuntu
- Microprocesador: Intel(R) Core(TM) i5 CPU M 480 @2.67 GHz (4 CPUs).
- Memoria RAM: 4GB.
- Disco Duro: 500 GB

A continuación, se describen las variables que miden el resultado de las pruebas realizadas al sistema:

Muestra: Cantidad de peticiones realizadas por cada URL.

Media: Tiempo promedio en milisegundos en el que se obtienen los resultados.

Mediana: Tiempo promedio en milisegundos en el que se obtuvo el resultado que ocupa la posición central.

Min: Tiempo mínimo que demora un hilo en acceder a la página.

Max: Tiempo máximo que demora un hilo en acceder a la página.

Línea 90%: Máximo tiempo utilizado por el 90% de la muestra, al resto de la misma le llevó más tiempo.

% Error: Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria.

Rendimiento (Rend): El rendimiento se mide en cantidad de solicitudes por segundo.

Kb/s: Velocidad de carga de las páginas.

Figura 17. Resultado de la prueba de rendimiento (Elaboración propia).

| Etiqueta | # Muestras | Media | Mediana | Línea de 90% | Mín | Máx | Rendimiento | Kb/sec |
|------------------|------------|-------|---------|--------------|-----|-------|-------------|--------|
| 332 /mena/nod... | 50 | 1359 | 1224 | 1542 | 505 | 3232 | 3,4/sec | 33,8 |
| 335 /mena/nod... | 50 | 1710 | 1690 | 2296 | 868 | 2522 | 3,4/sec | 30,6 |
| 323 /mena/nod... | 50 | 1664 | 1681 | 2211 | 782 | 3212 | 3,4/sec | 30,6 |
| 339 /mena/nod... | 50 | 1482 | 1384 | 2141 | 711 | 2782 | 3,4/sec | 30,9 |
| 337 /mena/nod... | 50 | 1276 | 1179 | 1730 | 613 | 2622 | 3,5/sec | 31,9 |
| Total | 250 | 2509 | 1629 | 7766 | 505 | 12431 | 13,3/sec | 120,7 |

3.3.3.1. Resultados de las pruebas de carga y estrés.

Se probó el sistema con un total de 250 peticiones enviadas en 50 hilos concurrentes, obteniéndose un tiempo promedio de respuesta de 3212 milisegundos (3.3 segundos aproximadamente), con un margen de error del 0.0 %.

3.3.4. Pruebas de integración

Las pruebas de integración son definidas para verificar el correcto ensamblaje entre los distintos módulos que conforman un sistema informático. Las mismas validan que estos componentes realmente funcionan juntos, son llamados correctamente y, además, transfieren los datos correctos en el tiempo preciso y por las vías de comunicación establecidas (Sommerville, 2005).

3.3.4.1. Resultados de las pruebas de integración

Partiendo de que la propuesta de solución es un módulo que debe integrarse al Sistema de Gestión Multieventos, se probó la integración del módulo que constituye la propuesta de solución de la presente investigación, al sistema mencionado anteriormente. Se pudo entonces constatar que los tipos de contenidos: solicitud de alimentación, solicitud de hospedaje, solicitud de locales, solicitud de materiales y solicitud de transporte, se crean correctamente. Las pruebas realizadas reflejan que el módulo de gestión

Módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI.



logística para el Sistema de Gestión Multieventos de la UCI, funciona correctamente y que sus funcionalidades no se vieron afectadas con la integración.

A continuación, en la **figura 18**, se muestra la interfaz gráfica, solicitud de materiales, donde se evidencia la integración del módulo al Sistema de Gestión Multieventos:

Panel de control | Contenido | Estructura | Apariencia | Usuarios | Módulos | Configuración | Informes | Ayuda | Bienvenido, admin | Cerrar sesión

Agregar contenido | Tipos de contenido | Traducir interfaz | Vistas | Rendimiento | Cron | Taxonomía | ADMIN

UCIENCIA/18

III Conferencia Científica Internacional
Universidad de las Ciencias Informáticas

CONVOCATORIA | TALLERES | COMITÉ | PROGRAMA | PAGOS Y HOSPEDAJE | SEDE | DESCARGAS

Inicio

| Ver | | Editar | | | | | | | |
|--|---------------------------------|--------------------------|------------------------|-------------------------|---|----------------------------|-----------------|----|----|
| Universidad de las Ciencias Informáticas | | UCI | | Solicitud de Materiales | | | D | M | A |
| Dirección: UCI | | Código: 123qwe | | | | | 1 | 11 | 19 |
| Almacén al que se solicita: 1 | | | Proceso: 3 | | | | | | |
| D e s t i n o | Orden de Producción No: 12 | | Lote No: 30 | | Centro de Costo: Dirección de compras y almacenes | | Código: 123qwee | | |
| | Orden de Trabajo No: Transporte | | Producto Código: 1123q | | Área: Subdirección de compras | | | | |
| Código | | Descripción | | U/M | | Cant. | | | |
| 1 | | Productos de carpintería | | kg | | 20 | | | |
| Solicitado | | | Autorizado | | | Recibido | | | |
| Nombre: Nadiela Milan Cristo | | | Nombre: Ramon morales | | | Nombre: Roynel Mena Chavez | | | |
| firma | | | firma | | | firma | | | |
| D M A | | | D M A | | | D M A | | | |
| 2 11 19 | | | 4 11 19 | | | 4 11 19 | | | |

(15): 14

Enviar ponencia

Figura 18. Interfaz gráfica Solicitud de Materiales (Elaboración propia).

3.3.5 Pruebas de aceptación

Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido. En esta prueba se evalúa el grado de calidad del software con relación a todos los aspectos relevantes para que el uso del producto se justifique (Software, 2016).

Luego de culminada la implementación del módulo de gestión logística para el Sistema de Gestión Multieventos, fue sometido a la prueba de aceptación por parte del cliente y de sus usuarios finales, la cual arrojó la completa satisfacción de estos, emitiendo así, la carta de aceptación del software.

3.4 Análisis de la aplicación de la técnica de ladov

Cuando se realiza una propuesta, es recomendable retroalimentarse con la opinión de los usuarios potenciales. Esta información es útil para conocer las debilidades de la propuesta y profundizar en sus fortalezas. Para valorar y/o constatar el nivel de aceptación y la aplicabilidad de la propuesta de solución se utiliza la técnica de ladov. Esta, es aplicada a una muestra de trabajadores y especialistas del Departamento de Ciencia, Tecnología e Innovación, para medir mediante el Índice de Satisfacción Grupal (ISG) el nivel de aceptación del sistema desarrollado, por parte de los usuarios finales.

Mediante esta técnica se puede determinar, de forma indirecta, Índice de Satisfacción Grupal (ISG) de los individuos involucrados en el proceso que está siendo objeto de análisis. Para la aplicación de la técnica de ladov se define un cuestionario conformado por cinco preguntas, de las cuales tres son cerradas y dos abiertas. Según (Ramírez, 2018), las preguntas cerradas guardan una relación entre sí, que previamente no es de conocimiento por parte del sujeto al que se le aplica la técnica.

Estas tres preguntas se relacionan a través del "Cuadro Lógico de ladov" el cual permite ubicar a cada encuestado en una escala de satisfacción, para luego calcular el ISG. Las respuestas a cada una de estas preguntas permiten determinar la posición de cada sujeto en la escala de satisfacción que toma valores desde 1 hasta 5, distribuido de la siguiente manera: 1-Clara satisfacción, 2-Más satisfecho que insatisfecho, 3-No definido o contradictorio, 4-Más insatisfecho que satisfecho y 5-Clara insatisfacción (Ramírez, 2018). El cuadro lógico utilizado en la investigación se muestra en la siguiente tabla:

Tabla 8. Cuadro Lógico de ladov utilizado (Elaboración propia).

| | | | | | | | | | |
|---|---|-------|----|-------|-------|----|----|-------|----|
| ¿Qué tan satisfecho se siente con los resultados del módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI? | ¿Estima provechoso el módulo desarrollado para el trabajo que se realiza en el departamento? | | | | | | | | |
| | Sí | | | No sé | | | No | | |
| | ¿Utilizaría el módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI? | | | | | | | | |
| | Si | No sé | No | Si | No sé | No | Si | No sé | No |



| | | | | | | | | | |
|---------------------------------|---|---|---|---|---|---|---|---|---|
| Clara satisfacción | 1 | 2 | 3 | 2 | 2 | 3 | 3 | 3 | 2 |
| Más satisfecho que insatisfecho | 2 | 2 | 3 | 2 | 3 | 3 | 2 | 3 | 2 |
| No definido o contradictorio | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| Más insatisfecho que satisfecho | 2 | 3 | 3 | 3 | 4 | 4 | 3 | 3 | 4 |
| Clara insatisfacción | 2 | 3 | 3 | 3 | 4 | 4 | 3 | 3 | 5 |

Para medir el grado de satisfacción de los usuarios respecto al sistema desarrollado, se tomó como muestra a siete especialistas y trabajadores vinculados a este proceso de gestión logística de eventos. La selección se realizó teniendo en cuenta la experiencia como analista en el trabajo con eventos (7 años como promedio) y el dominio de técnicas de análisis de gestión logística. Los resultados obtenidos para la satisfacción de forma individual se exponen a continuación.

A partir de la cantidad de respuestas por categoría es posible calcular el Índice de Satisfacción Grupal (ISG) siguiendo la siguiente fórmula:

$$ISG = \frac{A(+1) + B(+0.5) + C(0) + D(-0.5) + E(-1)}{N}$$

Las variables representan las cantidades de participantes agrupados por las escalas del índice de satisfacción individual. La cantidad de participantes que expresaron tener una clara satisfacción son representados por A, la cantidad que se sienten más satisfechos que insatisfechos se expresan mediante B, no definido o contradicción se evidencia mediante C, los que se sienten más insatisfechos que satisfechos mediante D, E es la cantidad de participantes que expresan una clara insatisfacción. El valor de N representa el total de participantes.

La escala numérica abarca el intervalo desde -1 hasta 1. Los valores que se encuentran comprendidos entre -1 y -0,5 indican insatisfacción; los comprendidos entre -0,49 y +0,49 evidencian contradicción y los que están entre 0,5 y 1 indican que existe satisfacción.

3.4.1. Resultados obtenidos de la aplicación de la técnica de ladov

1. Los resultados obtenidos de la aplicación de la encuesta se presentan a continuación:

Tabla 9. Escala del Índice de Satisfacción Individual (Elaboración propia).

Módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI.



| Categorías grupales de satisfacción | Escala | Participantes en la escala |
|-------------------------------------|--------|----------------------------|
| Clara satisfacción | A | 4 |
| Más satisfecho que insatisfecho | B | 3 |
| No definido o contradictorio | C | 0 |
| Más insatisfecho que satisfecho | D | 0 |
| Clara insatisfacción | E | 0 |

2. Cálculo del Índice de Satisfacción Grupal

$$ISG = \frac{A(+1) + B(+0.5)}{N}$$

$$ISG = \frac{4(+1) + 3(+0.5)}{7}$$

$$ISG = 0.79$$

El formulario presentado a los participantes incluye dos preguntas abiertas, mostradas a continuación:

1. ¿Qué importancia le concede al módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI?
2. ¿Qué aspectos a su juicio potencian o limitan el uso del módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI?

Sobre la primera pregunta, los participantes manifestaron que resulta de gran importancia porque permite realizar la gestión logística de los eventos gestionados por el Sistema de Gestión Multieventos, la cual se realizaba de forma manual, lo que permite agilizar el trabajo y procesar mayores volúmenes de información en menos tiempo; y, además, evita posibles pérdidas de información o duplicación de la misma.

Sobre la segunda pregunta los participantes expresaron que la creciente realización de eventos en la universidad, siempre va a requerir gestión logística, lo que potencia el uso de esta herramienta. En cuanto



a las limitaciones, consideran que todas se encuentran en el orden técnico, puede limitar su uso el despliegue del sistema en los servidores, su estabilidad y respaldo, constituye una posible limitante.

3. Interpretación del resultado del ISG

El proceso de validación de la propuesta de solución mediante la aplicación de una encuesta a especialistas en el análisis eventos y su gestión logística y el análisis de los resultados obtenidos mediante la técnica de ladov, confirmó su posibilidad de uso, expresado cuantitativamente en el alto Índice de Satisfacción Grupal (ISG = 0.79). Además, los criterios emitidos en las preguntas abiertas evidencian su satisfacción por la contribución del módulo al SGM, lo que refleja aceptación de la propuesta y un reconocimiento a su utilidad en función de su aporte práctico.

3.5. Conclusiones del capítulo

- La confección del diagrama de componentes permitió establecer una mayor comprensión de la arquitectura y funcionamiento de los elementos del módulo.
- La ejecución de pruebas al módulo para la gestión logística de los eventos académicos y científicos en el Sistema de Gestión Multieventos permitió detectar las deficiencias presentes, subsanarlas en el menor tiempo posible y ofrecer una aplicación con mayor calidad y seguridad.
- Como resultado de las pruebas funcionales, se detectaron en total, 10 no conformidades entre la primera y la segunda iteración, 8 de validación y 2 de redacción, las cuales fueron mitigadas con éxito, y en la tercera iteración, no se encontró ninguna no conformidad. Mediante las pruebas de integración se pudo constatar que el módulo de gestión logística funciona correctamente y que sus funcionalidades no se vieron afectadas con la integración. Y, mediante las pruebas de seguridad, se identificaron, en la primera iteración, 5 no conformidades, 1 alerta de riesgo medio, 2 de riesgo bajo y 2 notificaciones informativas, que fueron mitigadas con éxito.
- El proceso de validación de la propuesta de solución mediante la aplicación mediante la técnica de ladov, confirmó su posibilidad de uso, expresado cuantitativamente en el alto Índice de Satisfacción Grupal (ISG = 0.79). Además, los criterios emitidos en las preguntas abiertas evidencian su satisfacción por la contribución al Sistema de Gestión Multieventos, lo que refleja aceptación de la propuesta y un reconocimiento a su utilidad en función de su aporte práctico.



Conclusiones

En la presente investigación se realizó un proceso de desarrollo de software, con el objetivo de lograr un producto de calidad en el tiempo establecido. Una vez completada la investigación, se puede concluir que:

- El estudio de las soluciones informáticas existentes, relacionadas con la presente investigación, Open Conference Systems, EasyChair, El Sistema de Gestión para el Control de Actividades Científicas de los Docentes (SGCACD) y la Plataforma electrónica para la gestión de eventos científicos en la UCI, permitió identificar funcionalidades que sirvieron de base para el desarrollo de la solución, entre las que se encuentran exportar a pdf solicitudes creadas y envío de correos electrónicos.
- Como resultado del proceso de diseño se logró definir el ambiente de desarrollo, así como la metodología a utilizar, AUP en su variación UCI, que permitió la modelación de los artefactos que contribuyeron al diseño de la propuesta de solución y estandarizar el proceso de desarrollo del software.
- La obtención de los requisitos funcionales y no funcionales, garantizó que la solución respondiera a las necesidades del cliente y de los usuarios finales de crear, editar, eliminar, etc. Los tipos de contenido "Solicitud", sirviendo de guía para el desarrollo de las distintas funcionalidades del módulo.
- El módulo implementado mejora la gestión de eventos para la Dirección de Ciencia, Tecnología e Innovación, puesto que una vez informatizada dicha gestión, se eliminan los posibles casos de duplicación de información o pérdida de la misma y se realizan con mayor rapidez y eficiencia los procesos que requiere esta gestión.
- La aplicación de una estrategia de pruebas comprendida por pruebas funcionales, de seguridad, de rendimiento y de integración, permitió validar el correcto funcionamiento del sistema desarrollado, después de mitigar con éxito las diferentes no conformidades identificadas en cada una de las pruebas mencionadas anteriormente, y se constató la correcta integración del módulo en cuestión, al SGM, al comprobar que sus funcionalidades no se vieron afectadas con dicha integración.
- La validación científica a través de la técnica de ladov, después de comprobar los resultados obtenidos mediante el cuestionario conformado por las preguntas cerradas, el cálculo del Índice de

*Módulo de gestión logística para el Sistema de Gestión
Multieventos de la UCI.*



Satisfacción Grupal (ISG) y el formulario presentado con las preguntas abiertas, evidenció el nivel de aceptación del sistema por parte de los especialistas y trabajadores encuestados.



Recomendaciones

Para mejorar el módulo de la gestión logística para el Sistema de Gestión Multieventos de la UCI, se recomienda:

- Ampliar el concepto de gestión logística utilizado por la propuesta de solución, para así poder gestionar otros aspectos relacionados con esta gestión.



Referencias bibliográficas

1. **Acunetix. 2016.** Acunetix Web Vulnerability Scanner. [En línea] 2016. [Citado el: 13 de octubre de 2018.] <http://www.acunetix.com/vulnerability-scanner>.
2. **Alegsa, Leandro. 2010.** Alegsa.com.ar. [En línea] 12 de mayo de 2010. [Citado el: 11 de octubre de 2018.] <http://www.alegsa.com.ar/Dic/sghd.php>.
3. **Aleman Jimenez, Yulio y Thomas Sosa, Yoniel Jorge. 2014.** MÓDULO DE CONFIGURACIÓN PARA EL MECANISMO DE RASTREO DEL BUSCADOR ORIÓN. La Habana: s.n., 2014.
4. **Anderson, Michael, y otros. 2014.** An integrated development environment for faster feature engineering. 2014.
5. **Apache Software Foundation. 2016.** Apache Software Foundation. [En línea] 2016. [Citado el: 12 de octubre de 2018.] <http://jmeter.apache.org/>.
6. **Arredondo Morales, Perla Azucena. 2009.** Monografias. Servidores web. [En línea] 2009. [Citado el: 11 de octubre de 2018.] <http://www.monografias.com/trabajos75/servidores-web/servidoresweb.shtml>.
7. **Baydes Gonzáles, Eyllin. 2012.** Librería cliente para desarrollar aplicaciones con el marco de trabajo jWebSocket utilizando el lenguaje de programación Python. La Habana: s.n., 2012.
8. **Borrue, Julià. 2014.** Tutorial Drupal. [En línea] 2014. [Citado el: 3 de noviembre de 2018.] <http://www.cursosdrupal.com/content/arquitectura>.
9. **Brenji, Diego Javier. 2012.** Trabajo Final de Máster en Software Libre. [En línea] 2012. [Citado el: 7 de octubre de 2018.] <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/14890/4/brenjiTFM0612presentacion.pdf>.
10. **Brito, Eugenio González. 2016.** Pruebas de rendimiento de software. [En línea] 29 de 08 de 2016. [Citado el: 21 de octubre de 2019.]



11. **Burrows, William. 2010.** Visual Paradigm 8.0. [En línea] 16 de agosto de 2010. [Citado el: 8 de octubre de 2018.] <http://www.visual-paradigm.com/aboutus/newsreleases/vpuml80.jsp>.
12. **Cabrera Campos, Alcides. 2012.** El diseño metodológico de la investigación científica. La Habana : s.n., 2012. [Citado el: 8 de octubre de 2018.]
13. **Cardoso Said, Yadira y Barrera Garrido, Gladis De La Caridad. 2013.** Solución informática para gestionar la solicitud de participación de eventos científicos externos a la Universidad de las Ciencias Informáticas. La Habana: s.n., 2013.
14. **Cascading Style Sheets. 2013.** Guía Breve de CSS. [En línea] 2013. [Citado el: 11 de octubre de 2018.] <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>.
15. **Creed, Manuel. 2017.** Marco de Desarrollo de la Junta de Andalucía. [En línea] 03 de 04 de 2017. [Citado el: 10 de noviembre de 2018.]
16. **Drupal. 2016.** Coding standards | Drupal.org. [En línea] 2016. [Citado el: 12 de octubre de 2018.] <https://www.drupal.org/coding-standards>.
17. **Enamorado, Abel de la Rúa. 2014.** Comunidad virtual de Drupal. 2014.
18. **Ferrer, Laurato. 2017.** Propuesta de conceptualización de requisitos para proyectos software basados en formalismo de ingeniería de conocimiento. 2017.
19. **Fraser, Simon. 2014.** Public Knowledge Project. [En línea] 2014. [Citado el: 7 de octubre de 2018.] <http://pkp.sfu.ca/?q=ocs>.
20. **García Sánchez, Alexander. 2010.** Implementación del módulo de gestión de inventario. La Habana: s.n., 2010.
21. **Gauchat, Juan Diego. 2012.** El gran libro de HTML5, CSS3 y Javascript. Barcelona: s.n., 2012.
22. **Gómez, María del Carmen. 2015.** Notas del curso Base de Datos. Modelado de datos. MÉXICO, D.F: s.n., 2015.



23. **González, Liliana, y otros. 2015.** Campo de investigación en tecnologías de información y comunicación: estrategia de gobernanza en la Universidad de Medellín. Medellín: s.n., 2015.
24. **Márquez, Javier. 2013.** Patrones GoF. [En línea] Enero, de 2013. [Citado el: 7 de noviembre de 2018.] <https://infow.wordpress.com/category/patrones-de-disenogof/page/2/>.
25. **Martinez, A. 2016.** Metodologías para el desarrollo de software. [En línea]. 2016. [Citado: 11 de octubre de 2018] Disponible en: <http://es.slideshare.net/yeltsintorres18/metodologias-para-el-desarrollo-del-software>.
26. **Martínez Guaita, Alvaro. 2012.** Desarrolloweb.com. [En línea] 21 de febrero de 2012. [Citado: 11 de octubre de 2018] <http://www.desarrolloweb.com/actualidad/nuevo-apache-http-server-2-46573.html>.
27. **Netbeans. 2015.** Netbeans. [En línea] 2015. [Citado el: 11 de octubre de 2018.] <https://netbeans.org/community/releases/80/>.
28. **Nicolás, Bianco. 2015.** Consejo de Desarrollo Científico y Humanístico. [En línea] 2015. [Citado el: 2 de octubre de 2018.] <http://cdch-ucv.net/programa-de-financiamiento-para-laformacion/subprogramas/eventos-cientificos/>.
29. **Paulsen, Karl. 2011.** Moving Media Storage Technologies: Applications & Workflows for Video and Media Server Platforms. 2011. 978-0240814483.
30. **Pérez Reyes, Carlos Miguel y Argilagos Yi, José Miguel. 2012.** Plataforma electrónica para la gestión de eventos científicos en la Universidad de las Ciencias Informáticas. [En línea] [Citado el: 7 de octubre de 2018.] <http://publicaciones.uci.cu/index.php/SC/article/view/913>.
31. **Pressman, Roger. 2010.** Ingeniería de Software. Un enfoque práctico. 2010.
32. **Ramírez, Marlon. 2018.** Aplicación de la técnica de ladov. 2018.
33. **Real Academia Española. 2016.** Real Academia Española. [En línea] 2016. [Citado el: 2 de octubre de 2018.] <http://dle.rae.es/?id=TlvEXgq>.



34. **Rodriguez Sanchez. 2015.** Metodología de desarrollo para la Actividad productiva de la UCI v1.2. 2015, La Habana
35. **Rosabal, Antonio. 2011.** Sistema para el control de actividades científicas. Las Tunas: s.n., 2011.
36. **Salgueiro, Amado. 2001.** Indicadores de Gestión y Cuadro de Mando. 2001.
37. **Sarmiento, Johana. 2016.** Vision General de los diagramas de despliegue. [En línea] 22 de 03 de 2016. [Citado el: 13 de noviembre de 2018.]
38. **Silvente Lao, Yoel Joaquin. 2015.** Módulo para la detección de contenido duplicado en portales. 2015.
39. **Sommerville, Ian. 2005.** Ingeniería del software 7ma edición. 2005.
40. **Sommerville, Ian. 2011.** Ingeniería de Software. 2011.
41. **Software, Pruebas de. 2016.** Gestión de calidad y pruebas de software. <http://www.pruebasdesoftware.com/pruebadeaceptacion.htm>. [Online] 2016. [Citado el: 14 de noviembre de 2019.]
42. **Thibaud, Cyril. 2006. MySQL 5.** Barcelona: Amadeu Brugués, 2006. ISBN:2-7460-3069-1.
43. **Quimera, Felipe. 2015.** Patrones de diseño: qué son y por qué debes usarlos. [En línea] 2015. [Citado el: 7 de noviembre de 2018.] <http://www.genbetadev.com/metodologias-de-programacion/patrones-de-diseno-queson-y-por-que-debes-usarlos>
44. **Villamizar Suaza, Katerine. 2013.** Definición de equivalencias entre. Universidad Nacional de Colombia. Colombia: s.n., 2013.



Anexos

Anexo 1

Encuesta realizada a los especialistas de DOWAI:

1. ¿Qué tan satisfecho se siente con los resultados del módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI?

Clara satisfacción Más satisfecho que insatisfecho No definido o contradictorio

Más insatisfecho que satisfecho Clara insatisfacción

2. ¿Qué importancia le concede al módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI?

3. ¿Utilizaría el módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI?

Sí No sé No

4. ¿Qué aspectos a su juicio potencian o limitan el uso del módulo de gestión logística para el Sistema de Gestión Multieventos de la UCI?

5. ¿Estima provechoso el módulo desarrollado para el trabajo que se realiza en el departamento?

Sí No sé No