



Facultad 1

**Sistema para la gestión de medios de la Sede Cubana  
del Concurso Regional Caribeño del  
ACM-ICPC**

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

**Autor:**

Hector Daniel Rodríguez Lorenzo.

**Tutores:**

MSc. Aylin Estrada Velazco

Ing. Yordanka Fuentes Castillo

Ing. Liliana Martínez Sanabria

La Habana, junio de 2017



*Las cosas que nunca tuve, son tan sencillas  
como ir a buscar.*

*Pablo Milanés*

## DECLARACIÓN DE AUTORÍA

Yo Hector Daniel Rodríguez Lorenzo con CI: 93020211947 declaro ser único autor del presente trabajo de diploma y concedo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_

Firma del Autor

Hector Daniel Rodríguez Lorenzo

\_\_\_\_\_

Firma del Tutor

Ing. Yordanka Fuentes Castillo

\_\_\_\_\_

Firma del Tutor

Ing. Liliana Martínez Sanabria

\_\_\_\_\_

Firma del Tutor

MSc. Aylin Estrada Velazco

## AGRADECIMIENTOS

A Dios por su protección, su amor y su infinita misericordia.

A mis padres Jaquelin y Jorge Luis por darme la vida, por cuidarme, educarme y darme fuerzas a diario para ser mejor cada día y ser un orgullo para ellos.

A mi hermana Lizandra por apoyarme siempre y por ser mi alma gemela y lo más parecido a mí en cuestión de cromosomas.

A mis abuelos Martha, Virginia, Minino por su amor incondicional y Papío que aunque hoy no está, sé que está y estará siempre conmigo.

A Yoelvys que un día en el IPI hace 6 años y algo me dijo: “hay 10 plazas para la UCI vamos a apuntarnos a ver si tenemos suerte”.

A mis amigos Wendy, Arlenis y Lázaro (Mercedes) por soportarme todo este tiempo, incluso cuando ni yo me soportaba. A esos q vienen de antes y que son para siempre Yoandi, Yuri, Bety y Javielito, a los llegaron después pero que llegaron para quedarse Katuska, Grisel(Fifí), Jorge Eduardo y Oscar Luis.

A mis tíos, prim@s, padrinos y ahijadas por su preocupación y su apoyo.

A mis tutoras Yordanka, Aylin y Liliana por su guía y su ayuda en esta la recta final de la carrera.

A Liliana (Lili) por su amistad, su compañía, sus horas dedicadas, sus consejos, su preocupación y su afán por engordarme.

Al oponente, al tribunal de los cortes de tesis y la predefensa, por sus sugerencias y correcciones que me ayudaron a realizar un mejor trabajo.

A Yamir, Sergio, Dennys, Tio Felo y Martica por su apoyo y preocupación aún en la distancia.

A mis compañeros de aula, a los que estuvieron un tiempo y por razones x no están, gracias por su amistad y las historias compartidas.

A mis vecinos en especial Geidy, Eduardo y Mariloli por su apoyo, sus consejos y por ser parte de mi familia.

A mis profesores los cuales contribuyeron a mi formación como profesional.

A César y Alarcos Teatro por darme un lugar para hacer lo que más disfruto, por las enseñanzas y exigencias, por regalarme las mejores tardes entre ensayo y chuchuchu.

A mis amistades, a aquellas personas que estuvieron un día y hoy no están, a aquellos que me marcaron pero me hicieron a crecer y hacerme fuerte.

A la Universidad de las Ciencias Informáticas por ser mi casa, por ayudarme a madurar, a volverme independiente, por prepararme para la vida y darme la posibilidad de formarme como ingeniero, dirigente, artesano, modelo, declamador, actor, músico, poeta y loco.

A todos y todas, muchas gracias!!!

*A mis padres, mi hermana y mis  
abuelos por su amor y su confianza.*

## RESUMEN

Cuba constituye desde el año 2009 la sede regional caribeña del concurso internacional de programación competitiva ACM-ICPC. Por la importancia que posee este evento se hace imprescindible realizar un despliegue de recursos que garantice la calidad del evento. En la actualidad este despliegue de recursos se realiza de forma manual y auxiliado por documentos Excel, lo que ocasiona que se ralentice el proceso de asignación y devolución de los medios por los diferentes locales, así como la generación de reportes y estadísticas.

La solución propuesta permite llevar el control de los medios, ediciones del concurso, locales involucrados y miembros del staff, permitiendo un mejor manejo de la información y llevar a cabo de forma más organizada los procesos logísticos durante el evento. El proceso de desarrollo del *software* estuvo guiado por la metodología AUP-UCI, seleccionando para la implementación de la solución propuesta el CMS Drupal 7.54 y los lenguajes de programación PHP 5.6.25, CSS 3 y HTML 5 los cuales permitieron el desarrollo de tres módulos que brindan mejores funcionalidades al sistema. Luego de concluida la etapa de desarrollo se realizan un conjunto de pruebas que permiten detectar las no conformidades del sistema garantizando con su corrección un correcto funcionamiento del mismo.

Como resultado se obtuvo un sistema para la gestión de medios de la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC, que contribuye a informatizar el control y seguimiento de los mismos durante la realización del evento.

**Palabras claves:** ACM-ICPC, asignación, control, devolución, gestión, medios.

## ÍNDICE DE CONTENIDO

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>Capítulo 1: Fundamentación teórica de la investigación</b> .....	<b>8</b>
1.1 Gestión .....	8
1.1.1 Estudio de sistemas homólogos usados en el mundo: .....	9
1.1.2 Estudio de sistemas homólogos usados en Cuba: .....	10
1.2 Metodologías de desarrollo, herramientas y lenguajes .....	13
1.2.1 Metodologías de desarrollo de <i>software</i> .....	14
1.2.2 Herramientas .....	17
1.2.2.1 Sistema gestor de contenidos .....	17
1.2.2.2 Sistemas Gestores de bases de datos .....	19
1.2.2.3 Servidor de aplicaciones web .....	20
1.2.2.4 herramienta de modelado .....	22
1.2.2.5 Entorno de Desarrollo Integrado (IDE) de programación .....	22
1.2.2.6 Herramientas de validación .....	22
1.2.3 Lenguajes de Programación .....	23
1.2.3.1 Lenguajes del lado del cliente .....	23
1.2.3.2 Lenguaje del lado del servidor .....	24
1.2.3.3 Lenguaje de modelado .....	25
1.3 Conclusiones parciales .....	25
<b>Capítulo 2: Análisis y diseño del sistema de gestión de medios en la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC</b> .....	<b>26</b>
2.1 Descripción de la propuesta de solución .....	26
2.2 Modelo de dominio .....	26
2.3 Requisitos de la propuesta de solución .....	27
2.3.1 Requisitos funcionales .....	27
2.3.2 Requisitos no funcionales .....	29
2.4 Historias de Usuario .....	29
2.5 Descripción de la arquitectura de <i>software</i> y los patrones de diseño .....	32
2.5.1 Arquitectura de <i>software</i> .....	32
2.5.2 Patrones de diseño .....	33

2.6	Diagrama de clases del diseño .....	35
2.7	Diagrama de datos .....	38
2.8	Modelo de Despliegue .....	38
2.9	Conclusiones parciales .....	40
<b>Capítulo 3: Implementación y prueba del sistema de gestión de medios en la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC .....</b>		<b>41</b>
3.1	Diagrama de componentes .....	41
3.2	Estándares de codificación .....	42
3.3	Validación de la propuesta de solución .....	47
3.3.1	Pruebas de rendimiento (Carga y Estrés) .....	47
3.3.2	Pruebas funcionales .....	50
3.3.3	Pruebas de seguridad.....	53
3.3.4	Pruebas de integración .....	54
3.4	Validación de la hipótesis de la investigación .....	55
3.5	Conclusiones parciales .....	58
<b>CONCLUSIONES .....</b>		<b>59</b>
<b>RECOMENDACIONES.....</b>		<b>60</b>
<b>Referencias bibliográficas .....</b>		<b>61</b>
<b>Bibliografía .....</b>		<b>66</b>

## ÍNDICE DE TABLAS

Tabla 1 Operacionalización de las variables .....	4
Tabla 2 Comparación entre los sistemas homólogos estudiados .....	13
Tabla 3 Requisitos funcionales .....	27
Tabla 4 HU Añadir contenido medio AFT .....	30
Tabla 5 HU Editar contenido medio útil .....	31
Tabla 6 Resumen de los resultados de las pruebas de rendimiento .....	49
Tabla 7 Caso de prueba Añadir contenido medio AFT .....	50
Tabla 8 Caso de prueba Editar contenido medio útil .....	51
Tabla 9 Caso de prueba Asignar medio útil.....	51
Tabla 10 Resultados de las pruebas de seguridad.....	54
Tabla 11 Resultados de las pruebas de integración.....	55
Tabla 12 Expertos utilizados en la validación de la propuesta de solución.....	55
Tabla 13 Distribución de frecuencia para los datos primarios obtenidos .....	57

## ÍNDICE DE FIGURAS

Figura 1 Fases e Iteraciones de la variante de AUP para la UCI [12].....	16
Figura 2 Modelo de dominio.....	26
Figura 3 Prototipo Añadir contenido medio AFT.....	31
Figura 4 Arquitectura de la propuesta de solución.....	33
Figura 5 Uso del patrón Singleton.....	34
Figura 6 Uso del patrón Comand.....	34
Figura 7 Uso del patrón Chain of Responsibility.....	35
Figura 8 Uso del patrón Bridge.....	35
Figura 9 DCD con estereotipos web para Gestionar Medio_aft.....	36
Figura 10 DCD con estereotipos web para Gestionar Medio_util.....	37
Figura 11 Modelo de datos.....	38
Figura 12 Diagrama de despliegue de la propuesta de solución.....	39
Figura 13 Diagrama de componentes.....	41
Figura 14 Ejemplo de la apertura de etiquetas PHP.....	43
Figura 15 Ejemplo de indentación del código.....	43
Figura 16 Ejemplo del uso de operadores binarios.....	44
Figura 17 Ejemplo del uso de las comillas.....	44
Figura 18 Ejemplo de la estructura de control IF.....	45
Figura 19 Ejemplo del uso de la llave de apertura.....	45
Figura 20 Ejemplo del uso de llaves en estructuras de control.....	45
Figura 21 Ejemplo de la estructura ELSE y ELSE IF.....	45
Figura 22 Ejemplo del nombre de una función.....	46
Figura 23 Ejemplo de llamada a una función.....	46
Figura 24 Ejemplo del uso del espacio antes de una asignación.....	46
Figura 25 Ejemplo de un arreglo.....	46
Figura 26 Ejemplo de nombre de los archivos.....	47
Figura 27 Ejemplos de comentarios en el código.....	47
Figura 28 Resultado de las pruebas de Carga y Estrés.....	49
Figura 29 Comportamiento de las no conformidades por iteración.....	53
Figura 30 Comportamiento de la valoración de los expertos por categorías evaluadas.....	57

## INTRODUCCIÓN

El Concurso Internacional Universitario ACM-ICPC es un evento anual de programación y algorítmica entre universidades de todo el mundo, donde prima el trabajo en equipo, el análisis de problemas y el desarrollo rápido de *software*. Surgió en el año 1970 a partir de un concurso celebrado en la Universidad de Texas A&M (*Texas A&M University*). Pasó a ser una competición con varias rondas clasificatorias en 1977 y es en este momento cuando se realiza la primera Final Mundial. La Sede Central está ubicada en la Universidad de Baylor<sup>1</sup> desde 1989 y las competiciones regionales se ubican en universidades de todo el mundo, bajo el auspicio de la Asociación de Máquinas Computadoras (*Association for Computing Machinery*, ACM por sus siglas en inglés) y la colaboración de grandes empresas de la industria informática. La idea rápidamente ganó popularidad en los Estados Unidos y Canadá como iniciativa innovadora para elevar las aspiraciones, el rendimiento y la oportunidad de los mejores estudiantes en el campo de la Informática [1]. Desde entonces ha tenido un crecimiento acelerado en las cantidades de equipos, instituciones y países participantes. Solamente en el ciclo de competiciones 2016-2017, participaron 40 266 concursantes, agrupados en más de 128 equipos que representaron a 2736 instituciones de 102 países. Por su parte, a nivel mundial y desde el año 1997, el evento es patrocinado por la empresa IBM (*International Business Machines*) [2].

Estos concursos se realizan cada año en tres niveles de competición para clasificar a la Final Mundial del ACM-ICPC: [1]

- Concursos Locales (Nivel 1): se realizan a nivel de institución y tienen como propósito fundamental la clasificación de equipos para niveles superiores de competición.
- Concursos Nacionales (Nivel 2): se realizan a nivel de país y tienen como propósito fundamental la clasificación de equipos para el próximo nivel de competición.
- Concursos Regionales (Nivel 3): se realizan en varias regiones del mundo, entre octubre y diciembre de cada año. Tienen como propósito fundamental la clasificación de equipos para la Final Mundial del ACM-ICPC.
- Final Mundial: se realiza en los primeros meses del año siguiente a los Concursos Regionales y tiene como propósito fundamental la determinación de los mejores equipos a nivel mundial.

---

<sup>1</sup> Baylor University, Waco, Texas, EEUU

En el año 2009 Cuba se convierte en el primer país caribeño en ser una de las sedes del Concurso Regional del ACM-ICPC que abarcaba la zona de México, Centroamérica y el Caribe. Según el Director Ejecutivo del evento: *“Lo más importante de esta edición es que por fin se inicia el movimiento de concursos de programación ACM-ICPC en esta área, pues aunque este movimiento lleva más de 30 años, el Caribe estaba prácticamente ajeno”* [3].

La Sede Cubana del Concurso Regional Caribeño del ACM-ICPC pretende incrementar la participación de instituciones del Caribe y proveer prestigio, reconocimiento y visibilidad mundial para las instituciones cubanas y la Educación Superior en general [4]. Entre sus principales metas se destacan:

1. Alentar el desarrollo y reconocimiento de habilidades en: la programación de computadoras, las matemáticas, el idioma inglés, la resolución de problemas y el trabajo en equipo.
2. Proporcionar un espacio donde estudiantes y profesores puedan intercambiar culturas, experiencias y conocimientos.
3. Proveer una plataforma para orientar e incentivar la atención del público hacia la próxima generación de profesionales.
4. Clasificar equipos del Caribe para la Final Mundial del ACM-ICPC.

El Concurso Regional Caribeño es un evento de gran magnitud que abarca a todos los países del Caribe e involucra una gran cantidad de participantes. Cada año se incorporan más países, instituciones y equipos a estas competiciones caribeñas del ACM-ICPC, requiriendo un mayor esfuerzo por parte del Comité Directivo y apoyo del personal voluntario.

Desde el año 2009 se realizan en Cuba las competiciones regionales del ACM-ICPC teniendo como sede hasta la actualidad la Universidad de las Ciencias Informáticas (UCI). En la organización del evento, por la connotación e importancia que posee a nivel mundial, se tiene una cantidad significativa de procesos imprescindibles para su realización satisfactoria. En esta área un grupo de profesores, especialistas y estudiantes son los encargados de llevar a cabo la organización y realización de este concurso en sus diferentes niveles.

Uno de los procesos a llevar a cabo antes, durante y después de realizado el evento es la gestión de todos los recursos a utilizar o utilizados, ejemplos de estos pueden ser computadoras, cables de red, televisores, micrófonos, equipos de audio, banderas, globos, manteles entre muchos otros que en conjunto garantizan la calidad del evento. Para un mejor entendimiento de la presente investigación en el marco de la misma estos recursos serán tratados como “medios”. Los medios son facilitados por diversas

áreas y una vez finalizado el evento muchos de estos han de ser devueltos. Para ello se lleva en formato duro el control de todos los recibos y distribuciones por los diferentes locales donde se realiza la competencia u otros donde se lleven a cabo procesos que influyen directamente en la calidad del evento. El control del despliegue logístico de medios se realiza en la actualidad de forma manual, en formato duro y auxiliándose de tablas realizadas en documentos Excel, lo que dificulta el control y seguimiento de los medios disponibles, asignados y devueltos, ocasionando tardanzas en las entregas y devoluciones y dificultades para conocer la ubicación y disponibilidad de los mismos, trayendo consigo la pérdida de tiempo en la gestión de los medios y obstaculizando el proceso de generación de reportes y estadísticas.

La situación actual del control de medios en el Concurso Regional Caribeño del ACM-ICPC, descrita anteriormente, permite plantear el siguiente **problema de investigación**: ¿Cómo contribuir a informatizar el control y seguimiento de los medios a utilizar durante el evento en la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC?

Para la presente investigación se plantea como **objeto de estudio**: el proceso de gestión de medios, enmarcado en el **campo de acción**: el proceso de gestión de medios en la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC.

Dada la necesidad de informatizar y hacer más eficiente los procesos de distribución y control de medios de la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC se plantea como **objetivo general**: Desarrollar un sistema que permita la gestión de medios en la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC para informatizar el control y seguimiento de los mismos. Para lograr el cumplimiento del objetivo general se definen los siguientes **objetivos específicos**:

1. Definir el marco teórico conceptual de la investigación luego de un estudio de sistemas similares.
2. Analizar y seleccionar la metodología, herramientas y lenguajes de programación a utilizar en la implementación del sistema para la gestión de medios en la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC.
3. Definir los requisitos y funcionalidades del sistema para la gestión de medios en la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC.
4. Implementar las funcionalidades del sistema para la gestión de medios en la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC.
5. Validar el sistema desarrollado.

Luego de haber abordado los elementos fundamentales del área de la ciencia a incidir y objetivos primordiales, se formula la siguiente **hipótesis de investigación**: El desarrollo del sistema de gestión de medios para la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC, contribuirá a informatizar el control y seguimiento de los medios necesarios a utilizar en la realización del evento. Teniendo en cuenta la hipótesis planteada se define como **variable independiente**: sistema de gestión de medios para la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC. Como **variables dependientes** se especifican: informatizar el control de los medios necesarios a utilizar en la realización del evento e informatizar el seguimiento de los medios necesarios a utilizar en la realización del evento.

Tabla 1 Operacionalización de las variables

Tipo de variable	Variable	Dimensión	Conceptualización	Indicadores	Herramientas	Unidades de medida
<b>Variable Independiente</b>	Sistema de gestión de medios para la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC	Procesos y prácticas para la gestión de los medios	Capacidad de controlar las asignaciones y devoluciones de los medios vinculados al evento	Satisfacción del cliente	Encuesta de satisfacción	MA -Muy Adecuado BA – Bastante Adecuado A - Adecuado PA – Poco Adecuado I – Inadecuado
<b>Variable Dependiente</b>	Control de los medios necesarios	Procesos de control	Controlar la disponibilidad de los medios a utilizar	Tiempo de respuesta	Prueba de Carga y estrés	Milisegundos

	a utilizar en la realización del evento	Flujo de información sobre los medios	en el evento	Satisfacción del cliente	Encuesta de satisfacción	MA -Muy Adecuado BA – Bastante Adecuado A - Adecuado PA – Poco Adecuado I – Inadecuado
<b>Variable Dependiente</b>	Seguimiento de los medios necesarios a utilizar en la realización del evento	Locaciones del evento	Dar seguimiento al destino de cada medio a utilizar durante la realización del evento	Tiempo de respuesta	Prueba de Carga y estrés	Milisegundos
		Organización del evento		Satisfacción del cliente	Encuesta de satisfacción	MA -Muy Adecuado BA – Bastante Adecuado A - Adecuado PA – Poco Adecuado I – Inadecuado

Para dar cumplimiento a los objetivos específicos se definieron las siguientes **tareas de investigación**:

1. Realización de un estudio sobre los sistemas homólogos nacionales e internacionales existentes.
2. Selección de la metodología de desarrollo.

3. Selección de las tecnologías, herramientas y estándares que se necesitan para implementar la propuesta de solución.
4. Definición de los requisitos funcionales y no funcionales de la propuesta de solución.
5. Implementación de la propuesta de solución.
6. Documentación de las pruebas funcionales, de rendimiento y de seguridad.

### **Métodos de investigación científica**

Para el desarrollo de la presente investigación se hace primordial un correcto uso de los métodos de investigación científica, que permitan la obtención de la información necesaria referente al objeto de estudio. Los métodos teóricos permiten tener un conocimiento general sobre el estado actual del tema escogido, su evolución y conceptos generales mientras que los 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. Los métodos empleados se mencionan a continuación:

#### **Métodos teóricos**

- **Analítico sintético:** Permitirá durante la investigación analizar las teorías y documentos, facilitando la extracción de los elementos más importantes que se relacionan con el proceso de gestión de medios en el ACM-ICPC.
- **Histórico lógico:** Su objetivo en la investigación es constatar teóricamente cómo han evolucionado los sistemas de gestión de medios.
- **Modelación:** Permitirá crear abstracciones del objeto de investigación, a través de las que se realizarán los modelos que servirán de base para cada una de las fases del proceso de desarrollo, permitirá modelar o representar las características del sistema y relaciones entre objetos que intervienen en los procesos implementados por la propuesta de solución.

#### **Método empírico**

- **Entrevista:** La aplicación de este método posibilitó entender cómo se realiza actualmente el proceso de entrega y devolución de los medios en las competencias ACM-ICPC en la sede caribeña, además permitió identificar requisitos funcionales y no funcionales que debe cumplir el sistema.

El Trabajo de Diploma consta de introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos que ampliarán la información que se aporta en la investigación. A continuación, se describe brevemente la estructura del mismo, sintetizando el contenido de cada capítulo:

### **Capítulo 1: Fundamentación teórica de la investigación**

Se declara la base conceptual y se tratan los aspectos teóricos que dan sustento a la investigación, tales como: sistemas homólogos vinculados al campo de acción enunciado, las herramientas, tecnologías y metodologías de desarrollo adecuadas a la propuesta de solución.

### **Capítulo 2: Análisis y diseño del sistema de gestión de medios en la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC**

Se presenta una descripción detallada de los requisitos funcionales y no funcionales del sistema, propiciando una modelación adecuada para la propuesta realizada, determinando así las funcionalidades y características que se conciben para el sistema. Se especifica el patrón arquitectónico y los patrones de diseño que se utilizan y se presenta la propuesta de solución a partir de los artefactos generados como parte de la metodología seleccionada.

### **Capítulo 3: Implementación y prueba del sistema de gestión de medios en la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC**

Se exponen los aspectos relacionados con el proceso de implementación del sistema. Se describe la solución a través de los diagramas de componentes y de despliegue, elementos que se tuvieron en cuenta para el producto final. Se documenta el proceso de obtención de la solución y una vez concluida la fase de implementación se realizan las pruebas funcionales, de seguridad y de rendimiento, que validan si el sistema cumple con los requerimientos establecidos.

## **Capítulo 1: Fundamentación teórica de la investigación**

En el presente capítulo se declara la base conceptual que sustenta la investigación, se realiza un estudio de los sistemas homólogos, las herramientas y tecnologías con el objetivo de determinar las más factibles para implementar una solución informática que resuelva el problema de investigación.

### **1.1 Gestión**

Según el diccionario *Larousse* Ilustrado, la gestión es: Acción y efecto de administrar; planificación, organización, dirección y control de los recursos de una organización. Entre los diferentes tipos de gestión se encuentran: la ambiental, la tecnológica, la pública, de proyecto, de conocimiento, de la información, etc. La gestión tiene el fin de obtener el máximo beneficio posible; este beneficio puede ser económico o social, dependiendo esto de los fines que persiga la organización.

#### **Sistema de gestión**

Un sistema de gestión es una herramienta que permitirá optimizar recursos, reducir costes y mejorar la productividad en la empresa. Este instrumento de gestión reportará datos en tiempo real que permitirán tomar decisiones para corregir fallos y prevenir la aparición de gastos innecesarios.

Los sistemas de gestión están basados en normas internacionales que permiten controlar distintas facetas en una empresa, como la calidad de su producto o servicio, los impactos ambientales que pueda ocasionar, la seguridad y salud de los trabajadores, la responsabilidad social o la innovación [7].

#### **Sistemas para la gestión de medios**

Luego de la investigación realizada se pudo constatar que no se conoce un sistema que gestione el préstamo de medios teniendo en cuenta el control de las existencias en el almacén, el proceso de distribución y posteriormente el proceso de recepción de los mismos. Teniendo en cuenta que los movimientos de medios dentro de una misma institución se realizan a través del inventario e inventario según el diccionario de la Real Academia Española es un asiento de los bienes y demás cosas pertenecientes a una persona o comunidad, hecho con orden y precisión [8], se encuentran soluciones con características similares las cuales pueden ser analizadas para ver su semejanza con la solución propuesta.

### 1.1.1 Estudio de sistemas homólogos usados en el mundo:

#### **EZOfficeInventory**

EZOfficeInventory es una solución para la gestión de inventarios basado en la nube, diseñada para todo tipo de empresas. Ofrece escaneo de código de barras, gestión de proveedores, gestión de compras, seguimiento de inventarios y todo en una misma suite de herramientas pensadas en la productividad. La función de gestión de inventarios del *software* permite al usuario dar seguimiento de los niveles de inventario usando códigos QR<sup>2</sup> y etiquetas de código de barras. EZOfficeInventory cuenta con una aplicación para móviles con funcionalidad de escaneo de código de barras y código QR, lo cual permite a los usuarios registrar el inventario usando su tablet o smartphone. Una característica clave es su capacidad de seguimiento vía GPS<sup>3</sup>, lo cual permite al usuario la localización de todo ítem. Esta funcionalidad ayuda a dar seguimiento a todos sus activos en un mapa interactivo para identificar con facilidad cualquier artículo extraviado. A pesar de todas las cualidades que posee la herramienta, se debe destacar que no está desarrollada sobre una plataforma libre y necesita estar conectada a Internet para su correcto funcionamiento.

#### **Inventoria Inventory Software**

*Inventoria* está diseñada para ser tan intuitiva como sea posible, por lo que después de una instalación rápida, optimizará sus procesos de inventario en cuestión de minutos. Ofrece numerosas características útiles como la capacidad de añadir notas, URL<sup>4</sup> y fotografías para describir cada artículo en el inventario, escanea códigos de barras para agregar nuevos elementos, se integra con otros programas empresariales para mantener datos de inventario en todos los aspectos de su negocio, facilita la transferencia de existencias entre distintas ubicaciones, hace posible ver los niveles de inventario por ubicación, categoría o general, además almacena en el historial del producto cuando se recibieron o se vendieron artículos. Asigna niveles de acceso adecuados a los diferentes usuarios y permite a varios usuarios acceder al programa de administración de inventario. Posibilita el acceso al programa a través de la interfaz web desde dispositivos móviles. La versatilidad de este *software* para la administración de inventario hace que el mismo sea una herramienta muy potente pero no se puede dejar de destacar que no es un sistema desarrollado sobre una plataforma libre.

---

<sup>2</sup> Del inglés **Quick Response code**, código de respuesta rápida.

<sup>3</sup> Del inglés **Global Positioning System**, Sistema de Posicionamiento Global.

<sup>4</sup> Del inglés **Uniform Resource Locator**, es un identificador de recursos uniforme.

### ***Inventory Tracker Plus***

*Inventory Tracker Plus* un es *software* de gestión de inventario que facilita organizar el inventario y aparejado a esto dar seguimiento a la información de los clientes y vendedores. Utiliza su escáner de código de barras para ingresar números de inventario y agregar elementos de inventario a las facturas. Genera una lista rápida de artículos que necesitan ser reordenados y crea órdenes de compra que se pueden enviar por correo o por correo electrónico para esos artículos de inventario. Este *software* de inventario crea fácilmente facturas, hojas de embalaje, etiquetas de envío e informes de inventario, mantiene un registro de todos sus clientes, proveedores, vendedores y otros contactos, incluyendo direcciones de facturación y envío e historial de pedidos. Organiza y crea listas de tareas y citas en formatos de calendario diarios, semanales o mensuales. Este *software* de seguimiento de inventario maneja fácilmente devoluciones, pérdidas en el inventario, ajustes de auditoría, órdenes atrasadas, órdenes especiales y cantidades ilimitadas. Permite añadir un número ilimitado de fotos de sus artículos y ver un historial completo de artículos recibidos y vendidos. *Inventory Tracker Plus* puede traducir a cualquier idioma las facturas y los informes de inventario, incluye ayuda integrada, una guía del usuario y una visita guiada completa. Este *software* es perfecto para el seguimiento de activos por lo que constituye una buena opción para personas que están buscando un enfoque sencillo a la administración de inventario. A pesar de lo antes mencionado se debe enfatizar que no está desarrollado en una plataforma libre y tiene un costo de ciento noventa y nueve dólares.

#### **1.1.2 Estudio de sistemas homólogos usados en Cuba:**

##### **Versat Sarasola**

Versat Sarasola constituye el primer sistema integral de gestión de contabilidad certificado. Desarrollado para la gestión económica eficaz y fiable [9]. Este ofrece al usuario la posibilidad de contar con un instrumento seguro, rápido, eficaz y de fácil manejo para la planificación, control y el análisis de la gestión económica. Ha sido diseñado para ser utilizado en cualquier entidad y permite llevar el control y el registro contable individual de todos los hechos económicos que se originan en las estructuras internas de las mismas y obtener los estados financieros y análisis económicos y financieros en estos niveles. Actualmente es utilizado en Cuba en más de 8 000 entidades de 29 organismos. Este sistema integrado cuenta con un conjunto de 12 módulos entre los que se encuentran control de activos fijos y control de inventarios.

En el módulo de Control de Inventarios se definen formatos del clasificador de productos para lograr una uniformidad en el registro y la agregación de información en los reportes de salida, se conceptualizan los movimientos para lograr una información amplia sobre los orígenes y destinos de los recursos. Permite el control de las existencias y movimientos en diferentes monedas. Muestra el cuadro diario de cada uno de los almacenes por las diferentes cuentas. Ofrece la posibilidad de duplicar documentos para agilizar los pases de los mismos y lograr que un mismo documento se convierta en otro con solo adicionar un mínimo de información, realiza un control de las existencias y movimientos por custodios y se emiten diferentes reportes e información de utilidad para la correcta administración de los recursos materiales [9].

El módulo de Control de Activos Fijos recoge las operaciones normales que en esta actividad se realizan (altas, bajas, modificaciones de los medios). Posee un asistente para la configuración del subsistema, la tarea más compleja de este módulo [9].

El sistema Versat Sarasola es una aplicación de escritorio implementado en lenguaje de programación Delphi. Trabaja solamente sobre el sistema operativo *Windows* y tiene soporte para base de datos SQL<sup>5</sup> Server 2000. Además, el *software* tiene incluidos otros módulos que no son necesarios en la propuesta de solución lo que entorpece el proceso de gestión de los medios y hace más lenta la capacitación y ejecución del *software*.

### **Assets**

El Sistema de Gestión Integral (Assets), es un sistema multiusuario que se monta en una plataforma de servidores SQL, dividido en módulos económicos que trabajan en conjunto para el control de las actividades económica, financiera y contable sobre los medios materiales y financieros. El módulo inventario del Assets divide los medios en dos tipos de cuentas: activos fijos y útiles herramientas. Estas últimas son las que se utilizan para realizar las actividades de mantenimiento, talleres, almacenes, así como los equipos de protección física. Comprende entre otros, herramientas manuales, artículos de protección personal, utensilios de laboratorios, mini calculadoras y utensilios menores de cocina. Es un sistema integral modular concebido para el control de la actividad económica empresarial. Se emplea en la UCI para calcular las nóminas y controlar los recursos laborales de que dispone esta entidad. Este sistema está desarrollado sobre herramientas propietarias y constituye una aplicación de escritorio lo que ocasiona que el usuario deba instalar la aplicación en cada estación de trabajo.

---

<sup>5</sup> Del inglés ***Structured Query Language***, lenguaje de consulta estructurada

## **SISCONT5**

SISCONT5 es un sistema que se aviene a las definiciones y conceptos del Ministerio de la Industria Básica, aunque por las acciones contables financieras que permite puede ser utilizado en otras entidades nacionales [9]. Está formado por varios módulos dentro de los que se encuentra inventarios y activos fijos tangibles. Puede ser explotado en régimen mono usuario y multiusuario. Se define para mono entidad y multientidad, en esta última existe el control de su acceso para las entidades en un mismo equipo de cómputo como servidor.

El módulo de Inventario maneja toda la información referida al Submayor de Inventarios de la entidad, garantizando el cuadro permanente con las respectivas cuentas de la Contabilidad General. El sistema está preparado para controlar el saldo de cada material en dos monedas, a partir de los procedimientos vigentes en el país en cuanto a política monetaria [9].

Permite el tratamiento de las contabilizaciones de forma transaccional, por resúmenes diarios o de forma personalizada según se defina por parámetros. Incorpora tratamiento de lotes a los productos y dos tipos de valoración de las cuentas FIFO<sup>6</sup> y Promedio Ponderado.

Este *software* no está implementado sobre herramientas libres y es una aplicación de escritorio lo ocasiona que se deba instalar la aplicación en cada estación de trabajo.

## **Rodas XXI**

Sistema multiempresa y multiusuario creado por CITMATEL<sup>7</sup> para la automatización de la gestión empresarial [9]. Contiene diferentes módulos que pueden usarse integrados o independientes dentro de los que se encuentran activos fijos tangibles e inventarios.

El módulo de activos fijos tangibles de RODAS XXI permite tener un control detallado de los activos fijos la su entidad, realizando en el mismo momento que se registra un movimiento, su contabilización. Se pueden realizar todo tipo de operaciones de activos fijos con facilidad en el momento que se desee, generando el documento asociado al movimiento de que se trate de forma automática previa configuración del sistema para ello. Permite el control por separado de los activos fijos que se encuentran en almacén de los que se encuentran en explotación. Es posible además realizar ajustes a los activos fijos con facilidad mediante una opción que brinda el sistema [9].

---

<sup>6</sup> Del inglés *first in, first out* o *FIFO*, Primero en entrar, primero en salir.

<sup>7</sup> Empresa de Tecnologías y Servicios Telemáticos Avanzados

Este sistema no está desarrollado sobre tecnologías libres y es una aplicación de escritorio lo que ocasiona que el usuario deba instalar la aplicación en cada estación de trabajo.

### Valoración de los sistemas homólogos

Una vez analizados los sistemas homólogos se concluye que no resultan soluciones factibles para gestionar los medios de la sede caribeña del concurso regional del ACM-ICPC debido a que fueron desarrollados sobre plataformas de *software* propietario, lo que implica incrementos de gastos en licencias de uso y mantenimiento del *software*. Además, los sistemas internacionales utilizan tecnologías que no son asequibles para Cuba y algunos de ellos requieren de la conexión a internet para su funcionamiento. Los sistemas nacionales constituyen aplicaciones de escritorio lo que ocasiona que el usuario deba instalar la aplicación en cada estación de trabajo e impide que la aplicación pueda ser ejecutada desde dispositivos móviles. Es importante destacar que el estudio de estos sistemas evidenció la necesidad de desarrollar la propuesta de solución ya que ninguna de las soluciones estudiadas resuelve del todo la problemática existente y ayudó a una mejor identificación de los requisitos funcionales de la solución que se propone. En la Tabla 3 se muestra la comparación realizada entre estos sistemas estudiados:

Tabla 2 Comparación entre los sistemas homólogos estudiados

Sistema	Software libre	Aplicación web	Dependencia de Internet
<i>EZOfficeInventory</i>		X	X
<i>Inventoria Inventory Software</i>		X	
<i>Inventory Tracker Plus</i>		X	X
Versat Sarasola			
ASSETS			
SISCONT5			
Rodas XXI			

### 1.2 Metodologías de desarrollo, herramientas y lenguajes

En el desarrollo del Sistema para la gestión de medios de la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC se utilizarán varias tecnologías como lenguajes de programación, herramientas de modelado, gestores de bases de datos y servidores web. A continuación, se determina la metodología

de desarrollo que se utiliza para guiar el proceso de desarrollo del *software* y se realiza una descripción de las herramientas y lenguajes que serán usados para la implementación de la propuesta de solución.

### **1.2.1 Metodologías de 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 [10].

#### **Variación AUP-UCI**

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de *software* de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo:

- Desarrollo Dirigido por Pruebas (test driven development - TDD en inglés)
- Modelado ágil
- Gestión de Cambios ágil
- Refactorización de Base de Datos para mejorar la productividad [12].

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva.

#### **Fases AUP**

1. Inicio: El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
2. Elaboración: El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
3. Construcción: Durante la fase de construcción el sistema es desarrollado y probado en el ambiente de desarrollo.
4. Transición: El sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación para finalmente ser desplegado en los sistemas de producción [12].

#### **Variación de AUP para la UCI**

Al no existir una metodología de *software* universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea

configurable se realiza una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

Una metodología de desarrollo tiene entre sus objetivos aumentar la calidad del *software* que se produce, AUP UCI para ello se apoya en el Modelo CMMI-DEV v1.3<sup>8</sup> el cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora centrándose en el desarrollo de productos y servicios de calidad.

Con la adaptación de AUP se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos y se redujo a 1 la cantidad de metodologías que se usaban y de más de 20 roles en total que se definían se redujeron a 11 [12].

Esta metodología define 4 escenarios para el trabajo con requisitos teniendo en cuenta las características del equipo de proyecto y del negocio. Estos escenarios se muestran a continuación:

**Escenario No 1:** Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), similar a una llamada y respuesta respectivamente, donde la atención se centra en cómo el usuario va a utilizar el sistema. Es necesario que se tenga claro por el proyecto que los Casos de uso del negocio (CUN) muestran como los procesos son llevados a cabo por personas y los activos de la organización.

**Escenario No 2:** Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información.

**Escenario No 3:** Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad. Se debe tener presente que este escenario es muy conveniente si se desea representar una gran cantidad de niveles de detalles y la relaciones entre los procesos identificados.

---

<sup>8</sup> **Capability Maturity Model Integration** es un modelo para la mejora y evaluación de procesos para el desarrollo.

**Escenario No 4:** Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una historia de usuario (HU) no debe poseer demasiada información.

Las disciplinas definidas en la variación de AUP para la UCI son: modelado de negocio, requisitos, análisis y diseño, implementación, pruebas internas, pruebas de liberación y pruebas de aceptación. Estas disciplinas (desde Modelado de negocio hasta Pruebas de Aceptación) se desarrollan en la Fase de Ejecución, de ahí que en la misma se realicen Iteraciones y se obtengan resultados incrementales, ver Figura 1.

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 [12].

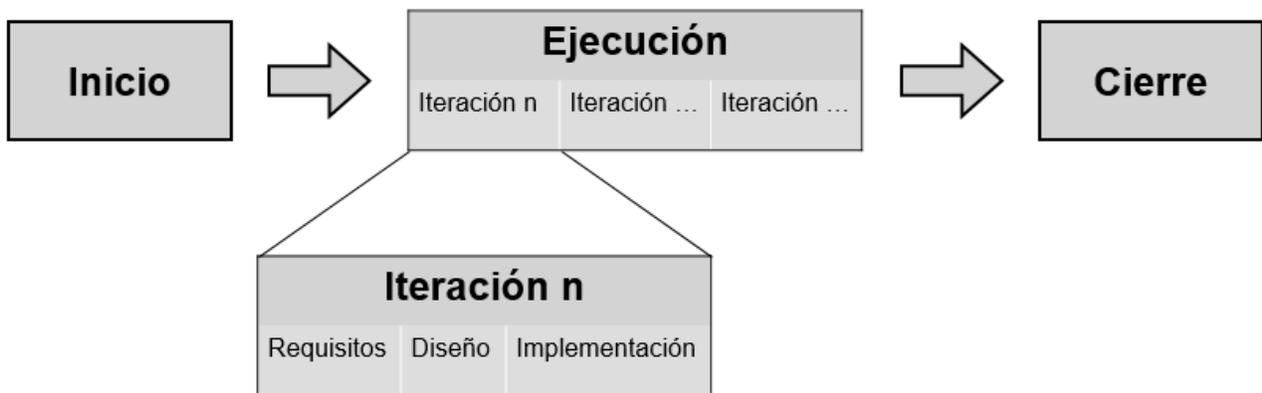


Figura 1 Fases e Iteraciones de la variante de AUP para la UCI [12].

### Metodología seleccionada

Para guiar el proceso de desarrollo de la propuesta de solución se decidió utilizar la metodología Variación AUP-UCI ya que es la definida por el proyecto, de esta forma se logra estandarizar el proceso de desarrollo de *software* dando cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3. Se decidió optar además por el escenario 4, el cual modela el sistema mediante historias de usuario.

## 1.2.2 Herramientas

Las herramientas informáticas son programas, aplicaciones o simplemente instrucciones usadas para efectuar tareas de modo más sencillo [13]. Con el objetivo de minimizar los costos, se propone utilizar tecnologías y herramientas que permitan su uso sin necesidad de pago de licencias. A continuación se hace un estudio para seleccionar las herramientas que se usarán en la solución propuesta.

### 1.2.2.1 Sistema gestor de contenidos

Un sistema de gestión de contenidos (*Content Management System* en inglés, abreviado CMS), es un *software* que permite crear una estructura base para la creación y administración de contenidos, principalmente de páginas web. Generalmente un CMS es una aplicación con una base de datos asociada en la que se almacenan los contenidos, separados de los estilos o diseño. El CMS controla también quién puede editar y visualizar los contenidos, convirtiéndose en una herramienta de gestión integral para la publicación de sitios web [14].

Las funcionalidades típicas de un CMS son:

- Administración de la estructura del portal: módulos, menús, diseño, configuración general. Administración del contenido: distintos tipos de contenidos, gestión y publicación de contenidos.
- Administración de usuarios: políticas de gestión de usuarios y de acceso a los contenidos mediante roles y permisos.
- Informes y gestión del portal: errores, estadísticas de acceso.

Algunos de los CMS genéricos utilizados en la actualidad son Wordpress, Joomla y Drupal, aunque existen diferencias notables entre ellos, comparten algunas características comunes ya que todos son *software* libre y gratuito, se programan en PHP y pueden correr en un servidor Apache. Teniendo en cuenta que el CMS que se utiliza en el Centro de Ideoinformática (CIDI) es Drupal en su versión 7 se hace un análisis de esta herramienta para elaborar la propuesta de solución.

### Sistema gestor de contenidos 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 [15].

El *software* está desarrollado con el lenguaje de programación PHP, utiliza una base de datos MySQL y está maquettato con hojas de estilo CSS. Drupal permite implementar un blog personal o profesional, portales corporativos, tiendas virtuales, redes sociales y comunidades virtuales [14].

#### **Características de Drupal:** [16]

- 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:** [17]

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 [16]. 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.

### 1.2.2.2 Sistemas Gestores de bases de datos

Un Sistema Gestor de Base de Datos (SGBD) se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos [18]. Para la selección del SGBD a utilizar fueron considerados PostgreSQL y MySQL.

**PostgreSQL:** Sistema Gestor de Bases de Datos objeto-relacional; basado en el proyecto POSTGRES, de la Universidad de Berkeley. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones y restricciones. Soporta los tipos base y otros de tipo como fecha, monetarios, elementos gráficos y cadenas de bits. También permite la creación de tipos propios. Incluye herencia entre tablas, por lo que se incluye entre los gestores de objetos-relacionales [19].

PostgreSQL se desempeña mejor en ambientes con altas cargas de usuario y consultas complejas y donde la integridad de los datos es muy importante. Una vez almacenados los datos, proporciona un sistema de respaldos en línea donde se puede ver una tabla en el estado en el que se encontraba en cierta fecha, proporcionando así un método flexible para la rápida recuperación de datos. Otra ventaja importante de este SGBD es la capacidad de su arquitectura para soportar módulos agregados.

**MySQL:** Sistema gestor de base de datos relacional, disponible para múltiples plataformas. Posee un licenciamiento dual, por un lado se ofrece bajo la Licencia Pública General de GNU (GNU-GPL

por sus siglas en inglés) para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. MySQL posee alta velocidad al realizar las operaciones, lo que le hace que posea un buen rendimiento [20]. Es considerado como de mediana escala y es adecuado para aplicaciones web de tamaño medio. MySQL permite que varios usuarios a través de múltiples hilos puedan acceder a datos relacionales y soporta múltiples herramientas para la gestión [21].

### **Selección del sistema gestor de base de datos**

Luego de valorar las opciones anteriores se decide usar MySQL como sistema gestor de base de datos en la propuesta de solución ya que su velocidad a la hora de realizar las operaciones, lo hace uno de los gestores que ofrecen mayor rendimiento, además su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.

#### **1.2.2.3 Servidor de aplicaciones web**

Un servidor web es un programa que utiliza HTTP (Protocolo de Transferencia de Hipertexto) para brindar los archivos que forman las páginas web a los usuarios, en respuesta a sus peticiones, que se envían por los clientes HTTP de sus equipos. El servidor web se encarga de interpretar el código PHP, generar la página HTML correspondiente y entregársela al usuario que la ha solicitado a través de su navegador. El proceso es un ejemplo del modelo de cliente / servidor. Todos los equipos que brinden servicios web deben tener programas de servidor Web. Los principales servidores web incluyen Apache, Internet Information Server de Microsoft (IIS) y Nginx. Otros servidores web incluyen servidor NetWare de Novell, Google web Server (GWS) y la familia de servidores Domino de IBM [22]. Para la selección del servidor de aplicaciones web a utilizar en la propuesta de solución fueron estudiados los siguientes:

#### **Nginx**

Nginx es un servidor web y proxy libre, de código abierto y de alto rendimiento, ofrece estabilidad, un gran conjunto de características básicas de los servidores web, configuración sencilla, y bajo consumo de recursos [23]. Nginx se integra con diferentes tecnologías web como Apache y PHP y presenta soporte de una amplia y activa comunidad [22]. Haciendo uso de sockets asíncronos utiliza un proceso por núcleo para manejar miles de conexiones, lo que permite un consumo de carga de la CPU y la memoria mucho más ligera [24].

## **Cherokee**

Cherokee es un servidor web de código abierto, rico en funciones, rápido y fácil de configurar, fue diseñado para la nueva generación de aplicaciones web seguras y altamente concurrentes. Ofrece soporte para las tecnologías más extendidas en la actualidad: FastCGI, SCGI, PHP, CGI, SSI, TLS y conexiones cifradas SSL, hosts virtuales, autenticación, codificación en tiempo real, balanceo de carga. Es altamente eficiente, extremadamente ligero y proporciona una sólida estabilidad. Entre sus muchas características hay una que merece un crédito especial: una interfaz amigable llamada *cherokee-admin* que se proporciona para una configuración sin complicaciones de cada característica del servidor. Esta interfaz de administración le permite configurar el servidor web sin tener que preocuparse por editar un archivo de texto escrito con una cierta sintaxis [25].

## **Apache**

Apache web Server es el servidor HTTP con mayor participación en el mercado mundial. Apache se caracteriza por ser estable, modular, tener código abierto y ser gratuito. Además, es altamente configurable de acuerdo a las necesidades de la organización que lo utilice. Apache registra en archivos *log* toda la operación con el fin de hacer más fácil la tarea del administrador suministrando información útil para la toma de decisiones, por ejemplo, el ajuste en una directiva de configuración. Por otra parte, Apache ofrece módulos especializados en distintas actividades, por ejemplo, integración con lenguajes de programación en el lado del cliente, módulos de seguridad y módulos de redirección. El uso de estos módulos puede ser aprovechado para dar un mejor servicio a sus usuarios sin comprometer a la organización que expone sus contenidos en una red pública como Internet para ser recuperados a través de solicitudes HTTP realizadas por un navegador. Apache se caracteriza también por ser multiplataforma, lo que permite su ejecución en la mayoría de sistemas operativos, tales como Unix, GNU/Linux y *Windows* [26].

## **Selección del servidor web**

Se selecciona el servidor web Apache por ser modular, multi-plataforma, extensible y de código abierto, por su capacidad de configuración, su potencia y por ser este servidor web recomendado para el desarrollo de portales y módulos en Drupal [16].

#### **1.2.2.4 herramienta de modelado**

**Visual Paradigm for UML 8.0** es una herramienta CASE<sup>9</sup> que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir diagramas de diversos tipos, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML [27]. Por las razones antes expuestas se emplea esta herramienta de modelado para la propuesta de solución.

#### **1.2.2.5 Entorno de Desarrollo Integrado (IDE) de programación**

##### **NetBeans 8.0**

NetBeans es un entorno de desarrollo integrado y una plataforma de desarrollo. Aunque inicialmente, NetBeans IDE sólo podía ser utilizado para desarrollar aplicaciones Java, a partir de la versión 6, NetBeans soporta varios lenguajes de programación mediante la instalación de complementos adicionales, algunos de estos lenguajes son Java, C, C ++, PHP, HTML, JavaScript, y Scala [28]. NetBeans proporciona funcionalidades adicionales como la compilación en tiempo real, la comprobación de tipos, refactorización, navegadores de clase y soluciones rápidas para los errores en tiempo de compilación.

#### **1.2.2.6 Herramientas de validación**

Las pruebas de validación en la ingeniería de *software* son el proceso de revisión que verifica que el sistema de *software* producido cumple con las especificaciones y que logra su cometido. La validación es una parte del proceso de pruebas de *software* de un proyecto, que también utiliza técnicas tales como evaluaciones, inspecciones y tutoriales. Es el proceso de comprobar que las especificaciones del usuario fueron cumplidas [29]. Para realizar las pruebas de validación a la solución propuesta se utilizan las herramientas Acunetix 8.0 y JMeter 3.1.

##### **Acunetix**

Acunetix es una herramienta de seguridad de aplicaciones Web automatizada. Es capaz de escanear cualquier sitio Web o aplicación Web que es accesible a través del protocolo HTTP / HTTPS. Acunetix

---

<sup>9</sup> Del Inglés: **Computer Aided Software Engineering**, Ingeniería de Software Asistida por Computadora

también proporciona herramientas de pruebas de penetración manuales que aumentan y contribuyen a las pruebas automatizadas, así como ayudar con la prueba de vulnerabilidades lógicas [30].

### **JMeter**

La aplicación JMeter de Apache es un *software* de código abierto, una aplicación Java 100% diseñada para cargar el comportamiento funcional y medir el rendimiento. Originalmente fue diseñado para probar aplicaciones web, pero desde entonces se ha expandido a otras funciones de prueba. Apache JMeter puede emplearse para probar el rendimiento en recursos estáticos y dinámicos, aplicaciones dinámicas en la Web. Se puede utilizar para simular una carga pesada en un servidor, grupo de servidores, red u objeto para probar su resistencia o para analizar el rendimiento general bajo diferentes tipos de carga. Las características de Apache JMeter incluyen: Capacidad para cargar y realizar pruebas de rendimiento en diferentes tipos de aplicaciones / servidor / protocolo. Fácil correlación mediante la capacidad de extraer datos de formatos de respuesta más populares, HTML, JSON, XML o cualquier formato textual, portabilidad completa, almacenamiento en caché y análisis *offline*/ reproducción de los resultados de las pruebas, núcleo altamente extensible [31].

### **1.2.3 Lenguajes de Programación**

Un lenguaje de programación consiste en un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Estos suelen usarse para crear programas que controlen el comportamiento físico y lógico de una máquina y para expresar algoritmos con precisión. A continuación, se describen los lenguajes de programación utilizados en el desarrollo de la propuesta de solución:

#### **1.2.3.1 Lenguajes del lado del cliente**

##### **Lenguaje HyperText Markup Language 5 (HTML)**

Lenguaje de publicación especificado como un estándar por el W3C (*World Wide Web Consortium*) que permite la creación de páginas web. Inicialmente fue presentado por Tim Berners-Lee que propuso un sistema basado en hipertexto para el intercambio de información en la Web. La aparición del lenguaje influyó notablemente en el crecimiento de Internet, donde la información era distribuida mediante colecciones fragmentadas de textos, imágenes y sonidos. HTML es independiente de la plataforma utilizada y se basa fundamentalmente en el uso de etiquetas estructurales y semánticas, adecuadas para la creación de documentos relativamente simples que permiten simplificar su estructura [32].

Algunas de sus características son:

- Estructura del cuerpo: permite agrupar todas estas partes de una web en nuevas etiquetas que representarán cada una de las partes típicas de una página.
- Etiquetas para contenido específico: utiliza etiquetas específicas para cada tipo de contenido en particular, como audio, video, etcétera.
- Bases de datos locales: permite el uso de una base de datos local, con la que se puede trabajar en una página web por medio del cliente y a través de un API<sup>10</sup>.
- Aplicaciones web *offline*: permite la creación de aplicaciones web que funcionen sin necesidad de estar conectados a internet.
- Geolocalización: permite localizar geográficamente las páginas web por medio de un API [32].

### **Lenguaje Cascading Style Sheets 3 (CSS)**

Con el crecimiento de Internet y la aparición del lenguaje HTML para la creación de páginas web, el W3C demostró la necesidad de un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos creados, de manera que pudieran visualizarse de igual forma en cualquier navegador web. A partir de entonces surgieron entre varias propuestas los lenguajes CHSS (*Cascading HTML Style Sheets*) y SSP (*Stream-based Style Sheet Proposal*), el primero realizado por Hakon Wium Lie y el segundo por Bert Bos, que a finales de 1994 y 1995 se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta denominado CSS (*Cascading Style Sheets*) [33].

CSS es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo utilizado para especificar el aspecto de una página web. Se basa en reglas que rigen el comportamiento del estilo de los elementos.

#### **1.2.3.2 Lenguaje del lado del servidor**

### **Lenguaje Hypertext Pre-processor 5 (PHP)**

Lenguaje de alto nivel con técnicas de Programación Orientada a Objetos, multiplataforma, sencillo de usar, rápido, integrable, además de ser de *software* libre, PHP es uno de los lenguajes de programación que permite programar *scripts* del lado del servidor, insertados dentro del código HTML, con una gran cantidad de funciones y mucha documentación. PHP es un lenguaje orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de

---

<sup>10</sup> Interfaz de programación de aplicaciones

datos. La programación en PHP es segura y confiable debido a que el código PHP es invisible al navegador y al cliente, el servidor es el encargado de ejecutar el código y enviar su resultado HTML al navegador [34].

Entre las características por las cuales se selecciona este lenguaje se encuentran:

- Drupal está implementado en PHP.
- Multiplataforma.
- Presenta soporte para bases de datos como: MySQL y PostgreSQL.

### **1.2.3.3 Lenguaje de modelado**

#### **Lenguaje Unificado de Modelado (UML)**

UML (Unified Modeling Language) fue adoptado como estándar del Object Management Group (Grupo Gestor de Objetos) en 1997 debido a que representa una colección de las mejores prácticas de ingeniería que han sido probadas con éxito en el modelado de sistemas. Es un lenguaje para la especificación, visualización, construcción y documentación de sistemas, no solo de *software*. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos informáticos, de arquitectura o de cualquier otra rama [35]. Por ser el lenguaje usado por la herramienta de modelado seleccionada, constituye UML el lenguaje de modelado escogido para la propuesta de solución.

### **1.3 Conclusiones parciales**

Como parte del desarrollo del presente capítulo se determinan las siguientes conclusiones parciales:

- Con el estudio de los referentes teóricos sobre el concurso del ACM-ICPC y su sede en el Caribe se evidenció la importancia que tiene esta competición para el mundo en el campo de la informática, lo que hace necesario una buena gestión de los medios que se destinan al evento con el objetivo de garantizar la calidad del mismo.
- El estudio de los sistemas homólogos demostró que no se cuenta con una aplicación que facilite la gestión de los medios en el concurso regional. Dada estas condiciones se evidenció la necesidad de desarrollar un sistema que permita la gestión de los medios que se destinan al concurso con el objetivo de informatizar dicha actividad.
- El estudio de las metodologías, herramientas y lenguajes permitió definir los componentes base para el desarrollo de la solución propuesta.

## Capítulo 2: Análisis y diseño del sistema de gestión de medios en la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC

En el presente capítulo se caracteriza la propuesta de solución a través de la definición de los requisitos funcionales (RF) y no funcionales (RNF), las tareas a realizar durante la implementación y la arquitectura de *software* y patrones de diseño utilizados en el desarrollo del sistema. Además, se presentan los principales artefactos generados en las primeras fases del desarrollo.

### 2.1 Descripción de la propuesta de solución

El Sistema de gestión de medios en la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC debe presentar funcionalidades que permitan gestionar la asignación y devolución de los medios empleados en el concurso, así como gestionar las ediciones, medios, locales y miembros del staff. El sistema brindará además la posibilidad de exportar los listados de medios, locales y miembros, así como notificar por correo luego de realizadas las asignaciones y devoluciones.

### 2.2 Modelo de dominio

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés [36].

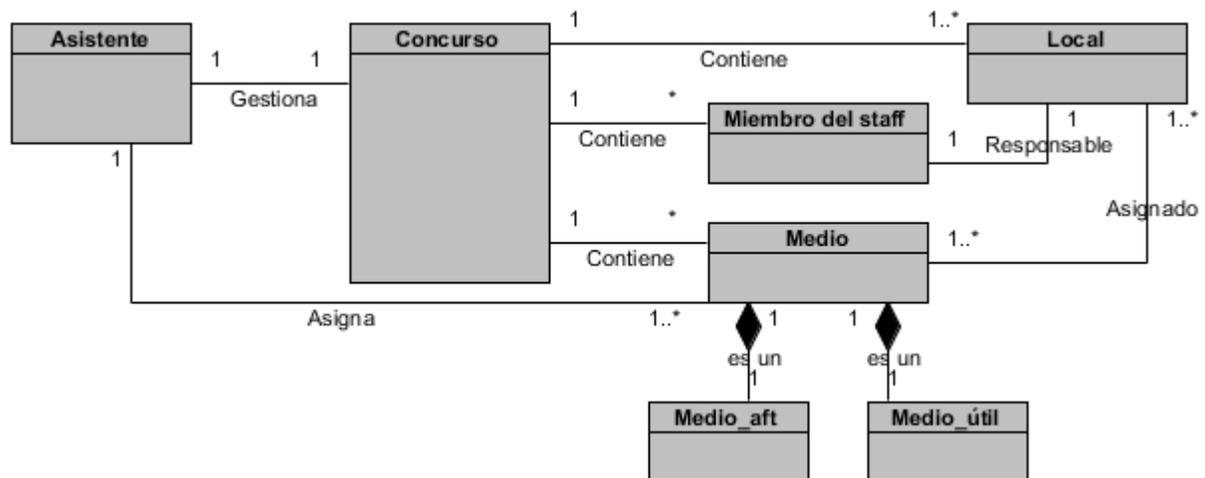


Figura 2 Modelo de dominio

## Descripción de los elementos del modelo de dominio

**Asistente:** Es el máximo encargado de la gestión del concurso y de garantizar la calidad del mismo. Visualiza y controla la información de la competencia y asigna los medios a los locales.

**Concurso:** Hace referencia al concurso regional del ACM-ICPC.

**Local:** Se refiere a los locales que estarán en función de la realización del evento.

**Miembro del staff:** Se refiere a los miembros del Staff, es decir, al personal que se encargará de llevar a cabo todos los procesos organizativos y logísticos de la competencia.

**Medio:** Hace referencia a todos los recursos que serán usados en la realización del evento.

**Medio\_aft:** Hace referencia a los recursos de tipo activo fijo tangible.

**Medio\_útil:** Hace referencia a los recursos de tipo útil.

### 2.3 Requisitos de la propuesta de solución

Los requerimientos para un *software* son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas [37]. Estos se dividen en:

#### 2.3.1 Requisitos funcionales

Los requisitos funcionales son declaraciones de las funcionalidades que debe cumplir el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares [29].

Tabla 3 Requisitos funcionales

Requisitos funcionales	Nombre del requisito funcional	Prioridad	Complejidad
RF1	Autenticar usuario	alta	baja
RF2	Añadir usuario	media	baja
RF3	Editar usuario	media	baja
RF4	Listar usuario	media	baja
RF5	Eliminar usuario	media	baja
RF6	Añadir contenido edición	alta	baja
RF7	Editar contenido edición	alta	baja

RF8	Mostrar contenido edición	alta	media
RF9	Añadir contenido medio AFT <sup>11</sup>	alta	baja
RF10	Editar contenido medio AFT	alta	baja
RF11	Mostrar contenido medio AFT	alta	media
RF12	Añadir contenido medio Útil	alta	baja
RF13	Editar contenido medio Útil	alta	baja
RF14	Mostrar contenido medio Útil	alta	media
RF15	Añadir contenido local	alta	baja
RF16	Editar contenido local	alta	baja
RF17	Mostrar contenido local	alta	media
RF18	Añadir contenido miembro	alta	baja
RF19	Editar contenido miembro	alta	baja
RF20	Mostrar contenido miembro	alta	media
RF21	Listar contenido	alta	baja
RF22	Eliminar contenido	alta	baja
RF23	Asignar medio útil	alta	alta
RF24	Devolver medio útil	alta	alta
RF25	Cambiar estado de medios AFT	alta	alta
RF26	Notificar acciones a responsables	alta	alta
RF27	Exportar vista a PDF	alta	media
RF28	Mostrar historial del medio AFT	alta	alta
RF29	Leer el identificador del medio AFT con lector de código de barras	media	baja

---

<sup>11</sup> AFT: Activo Fijo Tangible

### 2.3.2 Requisitos no funcionales

Los requerimientos no funcionales son requisitos que imponen restricciones en el diseño o implementación. Son propiedades o cualidades que el producto debe tener [34]. Clasificación según IEEE 810-1998 [38]:

#### 1. Requerimientos de interfaz externa:

RNF1: Predominar en el sistema los colores del logo del ACM-ICPC.

RNF2: Contener el logo del evento.

#### 2. Requerimientos de Performance:

RNF3: Soportar más de un usuario conectado concurrentemente.

#### 3. Restricciones de Diseño:

RNF4: Distribuir la arquitectura de la información en el panel izquierdo y superior garantizando el acceso a toda la información del sistema con no más de 3 clic.

RNF5: Requerir un servidor con 1Gb de RAM y un microprocesador Dual-Core como mínimo.

RNF6: Requerir un servidor web Apache en su versión 2.4.23 y PHP 5.6.25 o superior para el correcto funcionamiento de la interfaz web.

RNF7: Requerir un servidor de base de datos MySQL para hacer el sistema más ligero.

RNF8: Utilizar la licencia GNU/GPL versión 2 para el CMS Drupal.

RNF9: Utilizar la licencia PHP License.

RNF10: Utilizar la licencia de Visual Paradigm 8.

#### 4. Atributos del Sistema:

RNF11: Ajustar el diseño del sistema para al menos 5 resoluciones.

RNF12: Asignar permisos a los roles del sistema para limitar al acceso solo de personas autorizadas.

RNF13: Utilizar un proceso de encriptación irreversible para almacenar la contraseña de los usuarios.

### 2.4 Historias de Usuario

La Variación AUP-UCI utiliza en su escenario número 4 para describir los requisitos de *software* las Historias de Usuario, estas son descripciones cortas de las necesidades del cliente para el *software* que está en desarrollo [12]. Su utilización es común cuando se aplican marcos de trabajo ágiles [39]. A

continuación, se muestran las historias de usuario utilizadas en la propuesta de solución para los requisitos Añadir contenido medio AFT y Editar contenido medio útil.

Tabla 4 HU Añadir contenido medio AFT

<b>Número:</b> 9	<b>Nombre del requisito:</b> Añadir contenido medio AFT
<b>Programador:</b> Hector Daniel Rodríguez Lorenzo	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 24 horas
<b>Riesgo en Desarrollo:</b> ----	<b>Tiempo Real:</b> 24 horas
<p><b>Descripción:</b> Los usuarios que tengan rol administrador o editor podrán añadir nuevos contenidos de tipo medio en el sistema desde la vista de Administración, para ello deberán llenar los siguientes campos:</p> <ul style="list-style-type: none"> <li>• <b>Medio</b> (Obligatorio. Campo de texto. Permite solo los caracteres a-z, A-Z, números, guiones y espacio).</li> <li>• <b>Identificador</b> (Obligatorio. Campo de texto. Admite solo los caracteres N, S, M, B, guiones y números).</li> <li>• <b>Procedencia</b> (Obligatorio. Campo de autocompletamiento).</li> <li>• <b>Estado</b> (Obligatorio. Lista de selección. Muestra los valores (Disponible, Asignado, devuelto)).</li> <li>• <b>Descripción</b> (Opcional. Área de texto. Admite todos los caracteres).</li> </ul>	
<p><b>Observaciones:</b></p> <ol style="list-style-type: none"> <li>1. Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha creado satisfactoriamente el contenido.</li> <li>2. Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error.</li> <li>3. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben</li> </ol>	

llenarse.

**Prototipo elemental de interfaz gráfica de usuario:**

Crear medio AFT

Medio\*

Identificador\*

Procedencia\*

Estado\*

Descripción\*

Guardar Cancelar

Figura 3 Prototipo Añadir contenido medio AFT

Tabla 5 HU Editar contenido medio útil

<b>Número:</b> 13		<b>Nombre del requisito:</b> Editar contenido medio útil	
<b>Programador:</b> Hector Daniel Rodríguez Lorenzo		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 24 horas	
<b>Riesgo en Desarrollo:</b> ----		<b>Tiempo Real:</b> 24 horas	
<b>Descripción:</b> Los usuarios que tengan rol administrador o editor podrán editar contenidos de tipo medio en el sistema desde la vista de Administración, para ello deberán llenar los siguientes campos: <ul style="list-style-type: none"><li>• <b>Medio</b> (Obligatorio. Campo de texto. Permite solo los caracteres a-z, A-Z, números, guiones y espacio).</li><li>• <b>Cantidad</b> (Obligatorio. Campo de texto. Admite solo números).</li></ul>			

<ul style="list-style-type: none"> <li>• <b>Procedencia</b> (Obligatorio. Campo de autocompletamiento).</li> <li>• <b>Descripción</b> (Opcional. Área de texto. Admite todos los caracteres).</li> </ul>
<p><b>Observaciones:</b></p> <ol style="list-style-type: none"> <li>1. Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha creado satisfactoriamente el contenido.</li> <li>2. Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error.</li> <li>3. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben llenarse.</li> </ol>
<p><b>Prototipo elemental de interfaz gráfica de usuario:</b> No aplica</p>

## 2.5 Descripción de la arquitectura de *software* y los patrones de diseño

### 2.5.1 Arquitectura de *software*

Al utilizarse el CMS Drupal para el desarrollo de la propuesta de solución, la arquitectura de *software* a utilizar es la definida por el mismo, la cual es una arquitectura n-capas. Dividida específicamente en 5 capas que son descritas a continuación [40]:

**Plantillas (*templates*):** Esta capa establece la apariencia gráfica que se le muestra al usuario. Esta separación entre la información y los estilos permite cambiar la apariencia del portal web sin necesidad de modificar los contenidos. En la propuesta de solución esta capa contiene el tema *Premium responsive theme* aplicado al sistema, así como su codificación en el lenguaje CSS.

**Vistas (*views*):** Es la capa encargada de mostrar en los temas los cambios realizados a través de los módulos. Se evidencia esta capa en la propuesta de solución en las diferentes vistas (listado\_de\_medios\_aft, listado\_de\_medios\_útiles, listado\_de\_miembros, listado\_de\_ediciones, listado\_de\_locales, inventario\_de\_útiles).

**Entidades (*entities*):** Representa las entidades, que engloban los nodos, los usuarios, las taxonomías y los comentarios. Esta nueva estructura permite que sea posible añadirle campos a todo aquello que sea una entidad. En la propuesta de solución se engloban en esta capa los nodos referentes a los diferentes tipos de contenido (medio\_aft, medio\_útil, edición, local y miembro), los usuarios del sistema y las

taxonomías (área y procedencia) las mismas hacen referencia a las áreas de las cuales provienen los miembros del Staff y las áreas que destinan recursos para el desarrollo del concurso respectivamente.

**Módulos (*modules*):** Engloba los elementos que operan sobre los nodos otorgando funcionalidades a Drupal. Permiten incrementar sus capacidades o adaptarlas a las necesidades de cada portal web. Dentro de los módulos usados en la propuesta de solución, se encuentran: envío\_correo, asignar\_devolver\_aft y cálculo\_útiles\_disponibles, los cuales fueron implementados para lograr una mejor gestión de los medios.

**Base de Datos (*database*):** Esta capa es la encargada de gestionar el acceso a la información almacenada referente al funcionamiento del sistema y a los contenidos que serán mostrados a través del tema activo. La base de datos de la propuesta de solución tiene por nombre “medios4”.

Premium responsive theme	Plantillas
listado_de_medios_aft      listado_de_medios_útiles listado_de_miembros      listado_de_ediciones listado_de_locales      inventario_de_útiles	Vistas
medio_útil      local      área medio_aft      edición      procedencia miembro	Entidades
envío_correo asignar_devolver_aft cálculo_útiles_disponibles	Módulos
medios	Base de datos

Figura 4 Arquitectura de la propuesta de solución

### 2.5.2 Patrones de diseño

Los patrones de diseño expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas de *software*. Estos permiten diseñar sistemas seguros y que a su vez cumplan con los estándares de diseño establecidos por normas internacionales para el desarrollo de aplicaciones web [16]. Los patrones usados para el desarrollo del sistema se evidencian de la forma siguiente:

**Instancia única (*Singleton*):** Este patrón está diseñado para que solo permita que se haga una instancia de un elemento cuando este no existe. Dentro del núcleo de Drupal se utiliza este patrón de diseño pensando en los módulos y temas de Drupal como objetos para llevar a cabo la gestión de dichos

elementos, pues Drupal solo crea una instancia de un nodo, cuando este nodo no existe [16]. En el siguiente ejemplo se puede ver como el nombre del módulo coincide con el inicio de cada función, permitiendo que cada instancia se única, evitando que se repitan funcionalidades.

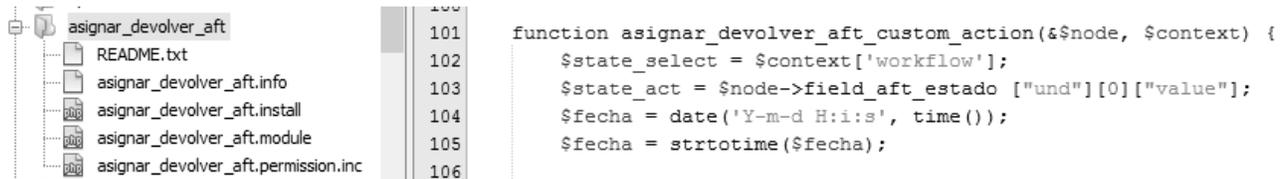


Figura 5 Uso del patrón *Singleton*

**Comandos (Comand):** Drupal utiliza este patrón para reducir el número de funciones que son necesarias para la aplicación, pasando la operación como un parámetro, junto con los argumentos y así agilizar la ejecución de ciertas tareas en el sistema [16]. El propio sistema de *hook* utiliza este patrón, por lo que los módulos no tienen que definir cada *hook*, sino más bien sólo los que se necesiten implementar.

```

function asignar_devolver_aft_custom_action_submit($form, $form_state) {
    $return = array();
    $return['workflow'] = $form_state['values']['workflow'];
    $return['local'] = $form_state['values']['local'];
    $return['entrega'] = $form_state['values']['entrega'];
    $return['recibe'] = $form_state['values']['recibe'];
    $return['devuelto_por'] = $form_state['values']['devuelto_por'];
    $return['aceptado_por'] = $form_state['values']['aceptado_por'];
    return $return;
}

```

Figura 6 Uso del patrón *Comand*

**Cadena de responsabilidades (Chain of Responsibility):** El sistema de menús de Drupal sigue este patrón. En cada solicitud de la página, el menú del sistema determina si hay un módulo para gestionar la solicitud y si el usuario tiene acceso a los recursos solicitados. Para ello, el mensaje se pasa a la opción del menú correspondiente a la vía de la solicitud. Si el elemento de menú no puede manejar la petición, se pasa a otro. Esto continúa hasta que un módulo se encarga de la petición, un módulo niega el acceso para el usuario, o la cadena se ha agotado [16]. La implementación del *hook\_menu* es un ejemplo claro que genera un menú de configuración del módulo.

```

function asignar_devolver_aft_menu() {
    $type = 'node';
    $items['node/%node/medios'] = array(
        'title' => 'Historial del medio',
        'page callback' => 'asignar_devolver_aft_tab_page',
        'page arguments' => array($type, 1),
        'access callback' => 'asignar_devolver_aft_tab_access',
        'access arguments' => array($type, 1),
        'type' => MENU_LOCAL_TASK,
    );
    return $items;
}

```

Figura 7 Uso del patrón *Chain of Responsibility*

**Puente (*Bridge*):** La capa de abstracción de bases de datos de Drupal se aplica de una forma similar a este patrón. Los módulos son escritos en una forma que es independiente del sistema que se está utilizando en la base de datos, y de esta forma, se puede añadir soporte para distintas bases de datos sin tener que modificar el código de un módulo [16].

```

if ($state_select == 1) {
    if (($state_act != "Asignado") && ($state_act != "Disponibile") ) {
        $node->field_aft_estado ["und"][0]["value"] = 'Disponibile';
        $query = db_insert('asignaciones_y_devoluciones_de_aft');
        $query->fields(array('nid', 'state', 'timestamp'));
        $query->values(array('nid' => $node->nid, 'state' => $context['workflow'], 'timestamp' => $fecha));
        $query->execute();
    }
}

```

Figura 8 Uso del patrón *Bridge*

## 2.6 Diagrama de clases del diseño

Un diagrama de clases de diseño (DCD) representa las especificaciones de las clases e interfaces *software* en una aplicación. A diferencia de las clases conceptuales del Modelo del Dominio, las clases de diseño de los DCD muestran las definiciones de las clases *software* en lugar de los conceptos del mundo real [36]. A continuación se muestra el diagrama de clases para gestionar medio AFT y gestionar Medio útil.

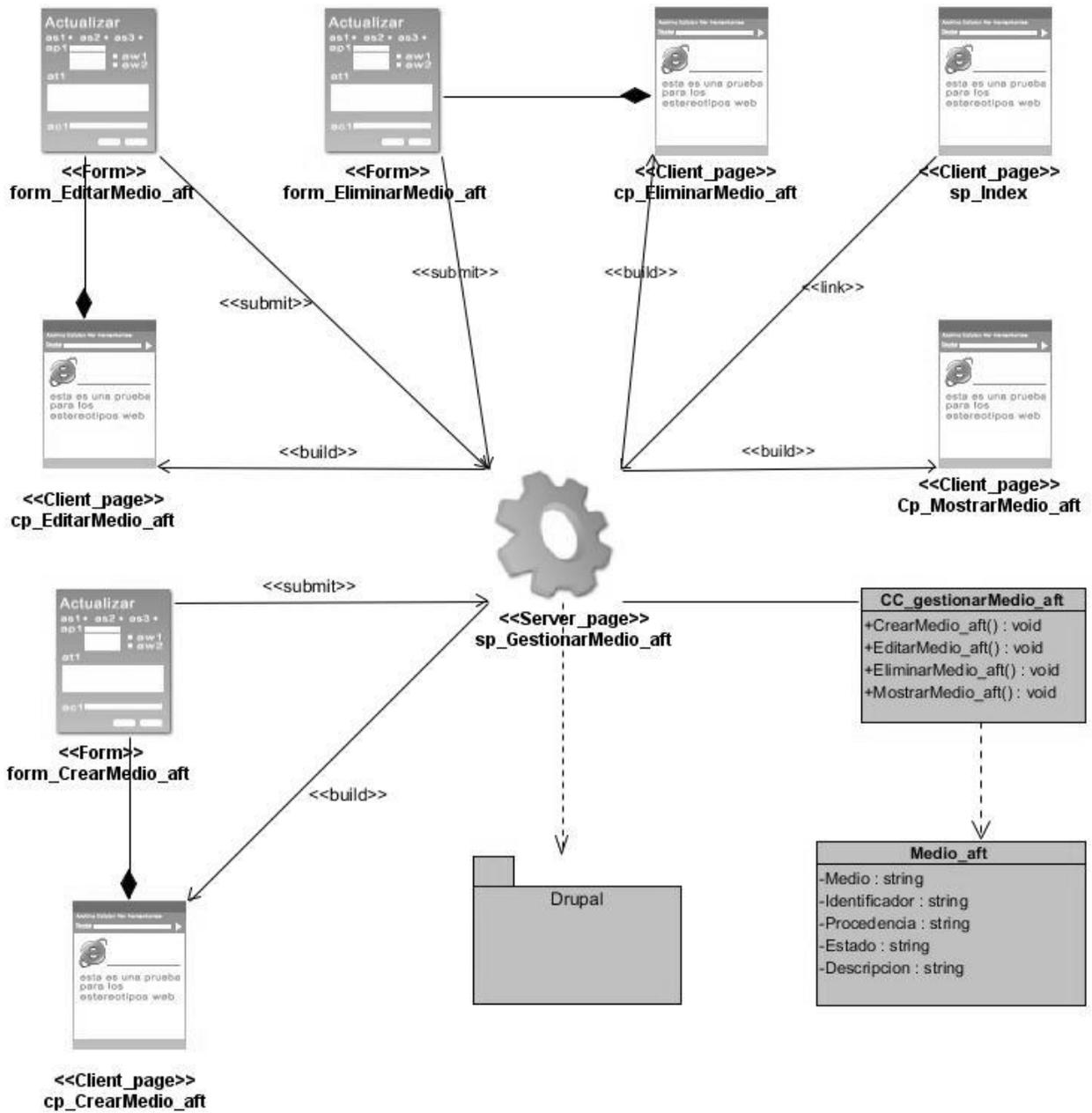


Figura 9 DCD con estereotipos web para Gestionar Medio\_apt

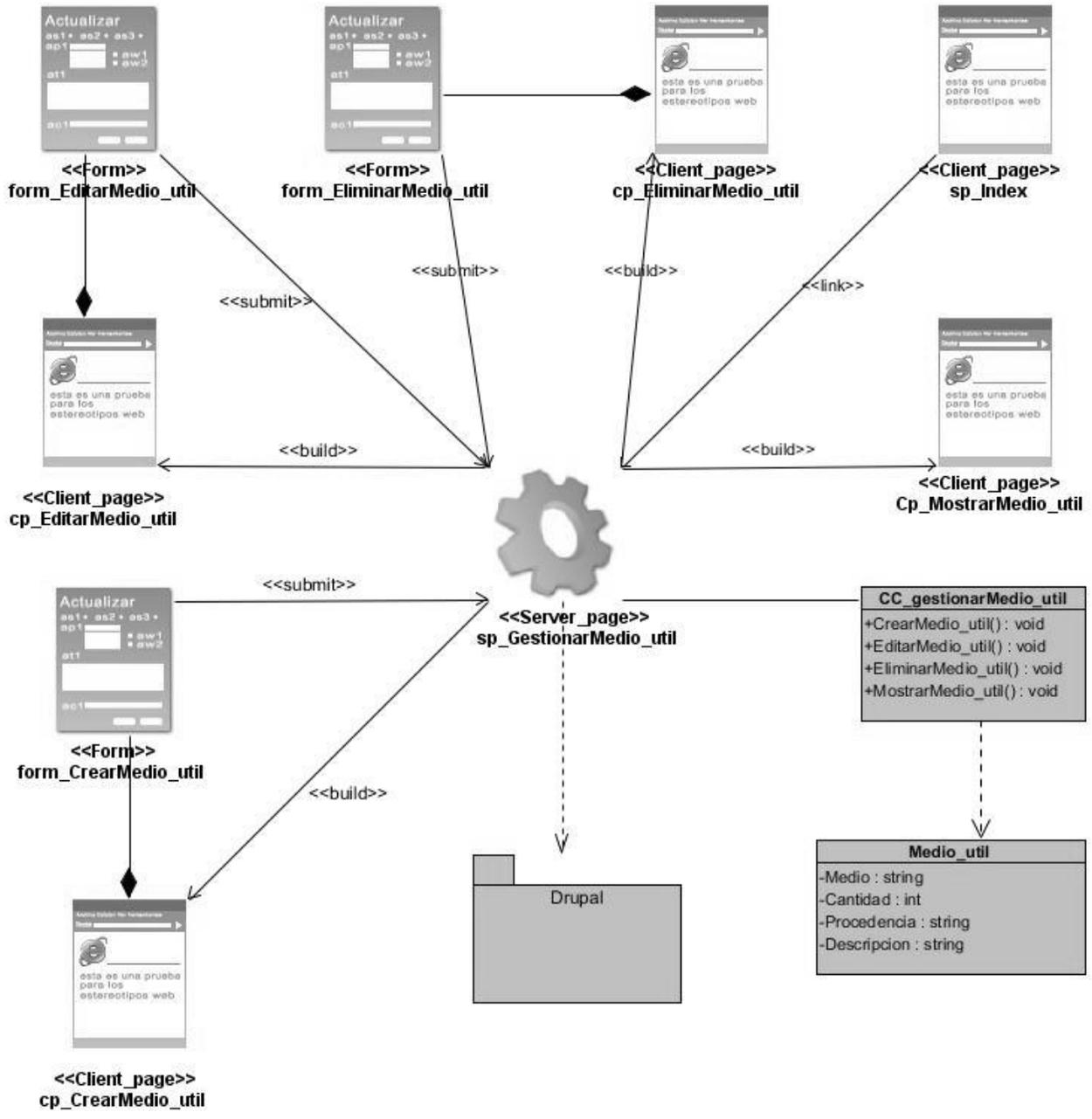


Figura 10 DCD con estereotipos web para Gestionar Medio\_util

## 2.7 Diagrama de datos

Un modelo de datos está orientado a representar los elementos que intervienen en la realidad o en un problema dado y la forma en que se relacionan dichos elementos entre sí [41].

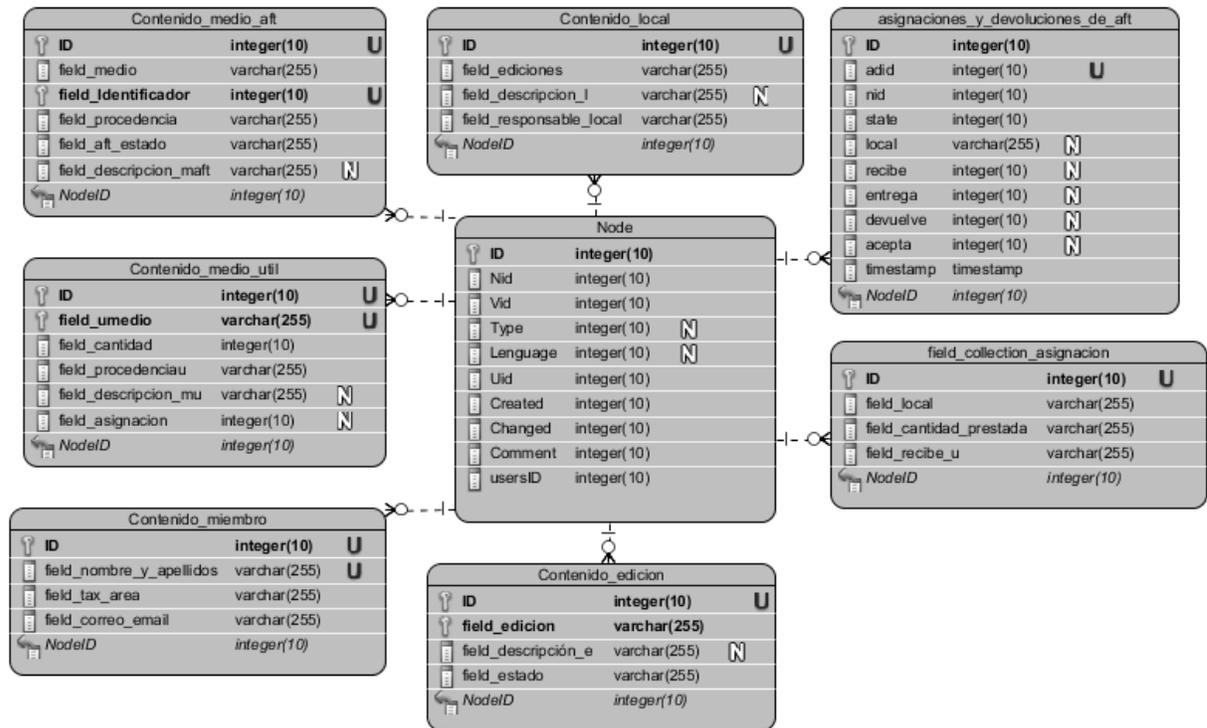


Figura 11 Modelo de datos

## 2.8 Modelo de Despliegue

Representa de forma visual las relaciones físicas que existen entre los componentes de *software* y hardware en el sistema. Los nodos son elementos de hardware sobre los cuales pueden ejecutarse los elementos de *software*. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño [42]. A continuación se presenta al modelo de despliegue correspondiente a la propuesta de solución.

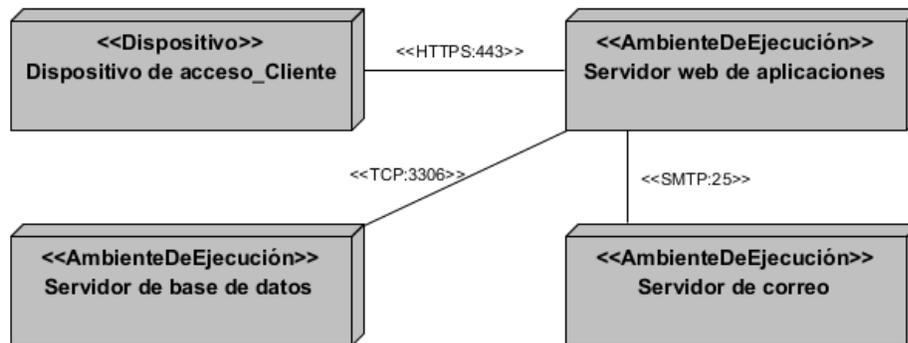


Figura 12 Diagrama de despliegue de la propuesta de solución

### Descripción de elementos e interfaces de comunicación:

**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.

## **2.9 Conclusiones parciales**

Como parte del desarrollo del presente capítulo se determinan las siguientes conclusiones parciales:

- Los requisitos funcionales y no funcionales identificados a partir del proceso de determinación de requisitos permitieron desarrollar las distintas funcionalidades que debe cumplir el sistema para solucionar las necesidades detectadas.
- Los artefactos que fueron generados de acuerdo a la metodología de desarrollo utilizada constituyeron una guía para la construcción de la propuesta de solución.
- Las actividades contempladas en el análisis y diseño, posibilitaron dar una mayor descripción de los elementos del sistema, permitiendo una mejor comprensión para su implementación.

### Capítulo 3: Implementación y prueba del sistema de gestión de medios en la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC

El presente capítulo presenta los componentes y estándares de codificación que sustentan la implementación del Sistema para la gestión de medios de la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC y se describe el proceso de validación de la solución implementada, mediante la utilización de los casos de pruebas.

#### 3.1 Diagrama de componentes

El diagrama de componentes muestra los componentes de un sistema de *software* conectados por las relaciones de dependencias lógicas entre cada uno de ellos. Provee una vista arquitectónica de alto nivel del sistema, ayudando a los desarrolladores a visualizar el camino de la implementación. Cada componente representa una unidad del código (fuente, binario o ejecutable), que permite mostrar las dependencias en tiempo de compilación y ejecución. La realización del diagrama posibilita tomar decisiones respecto a las tareas de implementación y los requisitos [43].

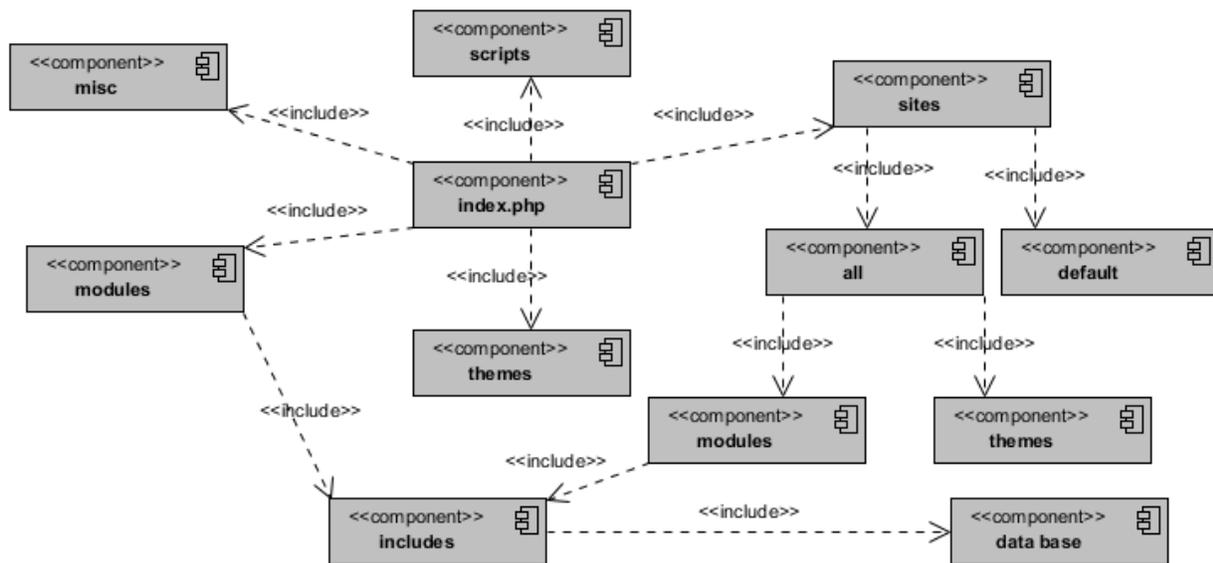


Figura 13 Diagrama de componentes

### **Descripción de los componentes del diagrama:**

**Index.php:** El archivo `index.php` es la página de inicio del sistema. A partir de esta entrada se solicitan los diferentes módulos del CMS Drupal 7.

**Themes:** Conjunto de temas que representan la forma en que se visualiza el sistema, los mismos vienen por defecto en el CMS Drupal 7.

**Misc:** Este directorio presenta un grupo de archivos necesarios para el sistema como: JavaScript y CSS e imágenes.

**Includes:** Este directorio es importante debido a sus funciones y *scripts* que posibilitan el funcionamiento de los módulos y del CMS Drupal, incluye además las API de conexión a la base de datos y que permiten la abstracción de Drupal a la base de datos empleada.

**Modules:** En esta carpeta se añaden los módulos que estarán disponibles durante la instalación. Generalmente sólo se usa esta carpeta para añadir módulos creados específicamente para la distribución. El resto de los módulos requeridos serán registrados como dependencias, de forma que se soliciten durante la instalación.

**Scripts:** Contiene utilidades adicionales que no utiliza Drupal directamente, pero que se puede utilizar desde la línea de comandos de *shell*.<sup>12</sup> Por ejemplo, el script `password-hash.sh` permite obtener una contraseña codificada a partir de la contraseña original (en texto plano).

**Sites:** La carpeta *sites* contiene los extras y modificaciones que se añaden a la distribución original. En esta se encuentran los módulos adicionales creados, añadidos o descargados del repositorio de módulos de Drupal y los temas adicionales instalados o creados. La carpeta incluye tras la instalación el archivo de configuración del sitio (*settings.php*).

### **3.2 Estándares de codificación**

Los estándares de codificación constituyen un principio esencial en el desarrollo de *software*. Garantizan que el código obtenido sea fácil de leer, entendido y modificado independientemente de quien haya sido el desarrollador del producto. Son una guía para el equipo de desarrollo, permiten asegurar que el código presente calidad y no contenga errores. Drupal proporciona a sus desarrolladores un conjunto de normas para fomentar el código de una forma uniforme para todos [16]. A continuación se detallan los estándares de codificación utilizados en la implementación de la solución propuesta.

---

<sup>12</sup> Hace referencia a Drush el cual es una herramienta que se instala en su servidor y que permite gestionar sus sitios en línea de comandos.

## Etiquetas de apertura y cierre PHP

Cuando se escribe en PHP, siempre se deben utilizar las etiquetas `<?php` y `?>`, y en ningún caso la versión corta `<?` y `?>`. En general se omite la etiqueta de cierre de PHP (`?>`) al final de los archivos `.module` y `.inc`. Esta convención evita que se puedan quedar olvidados espacios no deseados al final del archivo (después de la etiqueta de cierre `?>`), que serían identificados como salida HTML y podrían provocar un error muy típico, *“Cannot modify header information-headers already sent by...”*. Por tanto, la etiqueta de cierre final del archivo (`?>`) es opcional en Drupal. En la (Figura 14) se puede apreciar un fragmento de código el cual evidencia el uso de las etiquetas de apertura de PHP.

```
<?php
function asignar_devolver_aft_action_info() {
    return array(
        'asignar_devolver_aft_custom_action' => array(
            'type' => 'node',
```

Figura 14 Ejemplo de la apertura de etiquetas PHP

## Indentación

La Identación consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En programación se emplea Identación para anidar elementos. En Drupal se debe indentar con 2 espacios, nunca con tabuladores. Además no se debe dejar espacios en blanco al final de cada línea. En el siguiente ejemplo (Figura 15) se presenta un fragmento de código que demuestra la indentación del código.

```
function asignar_devolver_aft_action_info() {
    return array(
        'asignar_devolver_aft_custom_action' => array(
            'type' => 'node',
            'label' => t('Cambiar estados'),
            'behavior' => array('changes_property'),
            'configurable' => FALSE,
            'vbo_configurable' => TRUE,
            'triggers' => array('any'),
        ),
    );
}
```

Figura 15 Ejemplo de indentación del código

## Operadores

Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio. Por ejemplo, `$max_age = 452`, en el lugar de `$max_age=452`. Esto se aplica a operadores como `+`, `-`, `*`, `/`, `=`, `!=`, `>`, `<`, `.` (Concatenación de cadenas), `+=`, `-=`, etc. Los operadores unarios como `++`, `--` no deben tener separación. En la (Figura 17) se puede apreciar lo antes explicado en un ejemplo en el código de la solución.

```
if ($total - $valor < 0) {
    form_set_error('field_cantidad', t('La cantidad es muy grande.'));
}
```

Figura 16 Ejemplo del uso de operadores binarios

## Uso de comillas

Se utilizan tanto comillas simples como la ('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. En la (Figura 17) se evidencia el uso de las comillas en la codificación de la propuesta de solución.

```
if ($node->type=='local'){
    if (!preg_match("/^[áéíóúñÁÉÍÓÚÑ\\w\\s-]+$/", $node->title)) {
        form_set_error('title', t('No es valido el nombre del local.'));
    }
}
```

Figura 17 Ejemplo del uso de las comillas

## Uso de punto y coma (;) en código PHP

Aunque PHP permite escribir líneas de código individuales sin el terminador de línea (;), como por ejemplo `<?php print $title ?>`. En Drupal es siempre obligatorio: `<?php print $title; ?>`. Es importante señalar que el cierre de la etiqueta php es opcional.

## Estructuras de control

Con respecto a las estructuras de control, hay que tener en cuenta las siguientes normas:

- Debe haber un espacio entre el comando que define la estructura (*if*, *while*, *for*, etc) y el paréntesis de apertura. Esto es así para no confundir las estructuras de control con la nomenclatura de las funciones.

```

if ($state_select == 1) {
    if (($state_act != "Asignado") && ($state_act != "Disponible") ) {

```

Figura 18 Ejemplo de la estructura de control IF

- La llave de apertura ( { ) se situará en la misma línea que la definición de la estructura, separada por un espacio.

```

function asignar_devolver_aft_custom_action_submit($form, $form_state) {
    $return = array();
    $return['workflow'] = $form_state['values']['workflow'];

```

Figura 19 Ejemplo del uso de la llave de apertura

- Se recomienda usar siempre las llaves { } aun en los casos en que no sea obligatorio su uso (una sola "línea" de código dentro de la estructura de control).

```

if (is_numeric($delta)){
    $valor += $field['field_cantidad_prestada']['und'][0]['value'];
}

```

Figura 20 Ejemplo del uso de llaves en estructuras de control

- Las estructuras *else* y *elseif* se escribirán en la línea siguiente al cierre de la sentencia anterior.

```

if ($estado=='1'){
    $estado_actual='Disponible';
}
else if ($estado=='2'){
    $estado_actual='Asignado';
}
else{
    $estado_actual='Devuelto';
}

```

Figura 21 Ejemplo de la estructura ELSE y ELSE IF

## Funciones

Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guion bajo. Además, se debe incluir siempre como prefijo el nombre del módulo o tema, para evitar así duplicidad de funciones. 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.

```
function asignar_devolver_aft_tab_page($entity_type, $entity = NULL) {
```

Figura 22 Ejemplo del nombre de una función

En la llamada a la función se aplican las mismas reglas anteriores con respecto a los parámetros, como se muestra a continuación:

```
$miembro = node_load_multiple(array(), array('type' => 'miembro'));
```

Figura 23 Ejemplo de llamada a una función

Como excepción, es posible usar más de un espacio antes de una asignación (=) para mejorar la presentación, cuando se estén realizando varias asignaciones en bloque:

```
$node = node_load($field['field_recibe_u']['und'][0]['target_id']);  
$email = $node->field_correo_email['und'][0]['email'];
```

Figura 24 Ejemplo del uso del espacio antes de una asignación

## Arreglos (Arrays)

Los valores dentro de un *array* (o matriz) se deben separar por un espacio (después de la coma que los separa). El operador => debe separarse por un espacio a ambos lados. Cuando la línea de declaración del *array* supera los 80 caracteres, cada elemento se debe escribir en una única línea. En este último caso, la coma de separación del último elemento también se escribirá, aunque no existan más elementos. De esta forma se evitan errores al añadir nuevos elementos al vector.

```
$form['local'] = array(  
  '#type' => 'select',  
  '#title' => t('Local'),  
  '#options' => $nodos,  
  '#states' => array(  
    'visible' => array(':input[id="state"]' => array('value' => '2')),  
  ),  
);
```

Figura 25 Ejemplo de un arreglo

## Nombres de archivos

Los nombres de archivos deben escribirse siempre en minúscula. La única excepción son los archivos de documentación, que tendrán extensión .txt y el nombre en mayúscula. Por ejemplo README.txt.

Nombre	Fecha de modifica...	Tipo	Tamaño
asignar_devolver_aft.info	26/04/2017 10:39 ...	Archivo INFO	1 KB
asignar_devolver_aft.install	06/05/2017 1:02 a...	Archivo INSTALL	2 KB
asignar_devolver_aft.module	05/05/2017 4:38 p...	Archivo MODULE	12 KB
asignar_devolver_aft.permission.inc	26/04/2017 11:49 ...	Archivo INC	2 KB
README	07/05/2017 4:28 p...	Documento de tex...	0 KB

Figura 26 Ejemplo de nombre de los archivos

### Comentar el código

Para la realización de comentarios suelen emplear */\** para comentarios en varias líneas y *//* para comentarios de una única línea. Se deben escribir frases completas, comenzándolas con mayúscula y terminándolas con un punto. En caso de que en el comentario se haga referencia a una constante, esta deberá escribirse en mayúscula (por ejemplo: TRUE o FALSE).

```
<?php
/**
 * Trabajo para la gestion de medios.
 */

//Funcion para validar cantidades de utiles
function calculo_utiles_disponibles_node_validate($node, $form, &$form_state) {
```

Figura 27 Ejemplos de comentarios en el código

### 3.3 Validación de la propuesta de solución

El único instrumento adecuado para determinar el status de la calidad de un producto de *software* es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del *software* o al sistema de *software* en su totalidad, con el objetivo de medir el grado en que cumple con los requerimientos [44].

#### 3.3.1 Pruebas de rendimiento (Carga y Estrés)

Las pruebas de rendimiento se diseñan para asegurar que el sistema pueda procesar su carga esperada. Éstas se ocupan tanto de demostrar que el sistema satisface sus requerimientos, como de descubrir problemas y defectos en el sistema [37].

Las pruebas de carga consisten en simular una carga de trabajo similar y superior a la que tendrá cuando el sitio esté funcionando, con el fin de detectar si el *software* instalado (programas y aplicaciones) cumple

con los requerimientos de muchos usuarios simultáneos y también si el *hardware* (servidor y el equipamiento computacional de redes y enlace que lo conecta a Internet) es capaz de soportar la cantidad de visitas esperadas [45].

Las pruebas de estrés evalúan la robustez y la confiabilidad del *software* sometiéndolo a condiciones de uso extremas. Entre estas condiciones se incluyen el envío excesivo de peticiones y la ejecución en condiciones de hardware limitadas. El objetivo es saturar el programa hasta un punto de quiebre donde aparezcan defectos potencialmente peligrosos [46].

### **Resultados de las pruebas de rendimiento**

Para las pruebas de rendimiento se utiliza el *software* Apache Jmeter v3.1. Para ello se definen las propiedades de las PC implicadas.

#### **Hardware de prueba (PC cliente):**

- Tipo de procesador: Intel(R) Core(TM) 2 Duo CPU P8400 @2.26GHz 2.27GHz
- RAM: 3 GB DDR2.
- Tipo de Red: *Ethernet* 10/100Mbps.

#### **Hardware de prueba (PC servidor):**

- Tipo de procesador: Intel(R) Core(TM) 2 Duo CPU P8400 @2.26GHz 2.27GHz
- RAM: 3 GB DDR2.
- Tipo de Red: *Ethernet* 10/100Mbps.

#### **Software instalado en ambas PC:**

- Tipo de servidor web: Apache 2.4.
- Plataforma: SO Linux (PC servidor) y SO *Windows* (PC cliente).
- Servidor de BD: *MySQL* 5.7.

Luego de definido el hardware se configuran los parámetros del Apache JMeter logrando un ambiente de simulación con un total de 5 usuarios conectados concurrentemente, se realizan peticiones a diferentes páginas del Sistema para la Gestión de medios de la sede cubana del concurso regional caribeño del ACM-ICPC. En la (Figura 28) se puede observar los resultados obtenidos por el sistema.

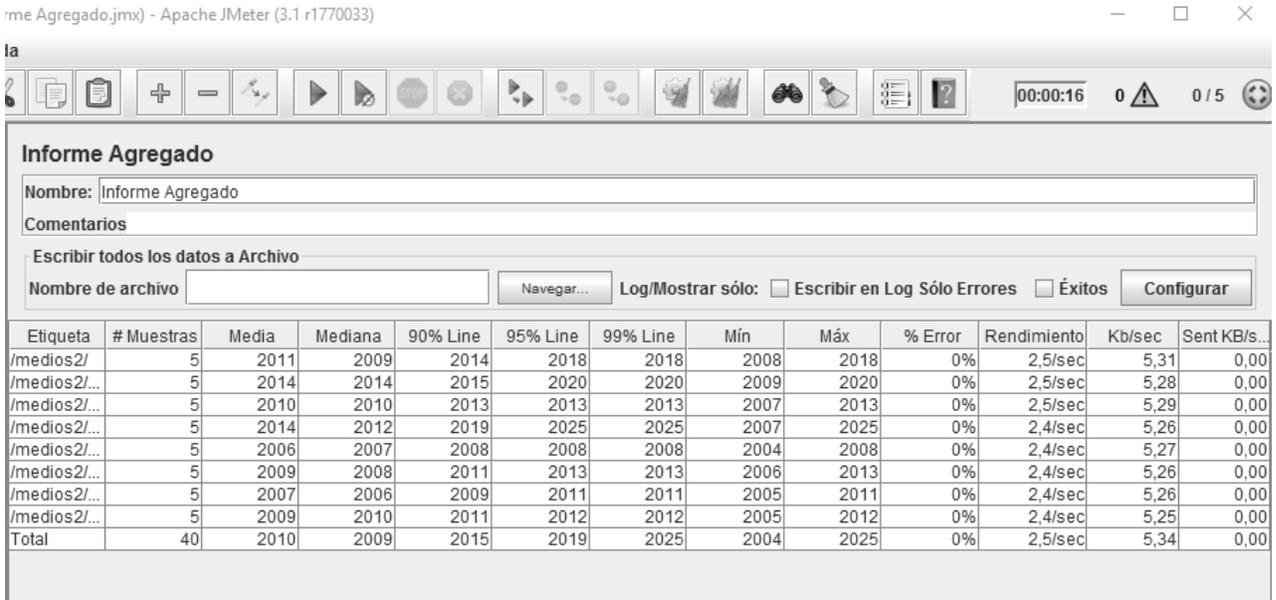


Figura 28 Resultado de las pruebas de Carga y Estrés

### Análisis de los resultados de las pruebas de rendimiento

Para un mejor entendimiento de las pruebas de Rendimiento, se explica cada parámetro que la compone a continuación:

- **Usuarios:** total de usuarios.
- **# Muestras:** El número de peticiones.
- **Media:** El tiempo medio transcurrido en milisegundos para un conjunto de resultados.
- **Mín:** El mínimo tiempo transcurrido en milisegundos para las muestras de la URL dada.
- **Máx:** El máximo tiempo transcurrido en un milisegundo para las muestras de la URL dada.
- **% Error:** Porcentaje de las peticiones con errores.
- **Rendimiento:** Rendimiento medido en base a peticiones por segundo/minuto/hora.
- **Kb/s Recibidos:** Rendimiento medido en Kbytes por segundos.

Tabla 6 Resumen de los resultados de las pruebas de rendimiento

Usuarios	# Muestras	Media	Mín	Máx	% Error	Rendimiento (peticiones/segundos)	Kb/s Recibidos
5	40	2010	2004	2025	0 %	2.5	5.34

El sistema desarrollado, para un total de 5 usuarios conectados de forma concurrente respondió 40 peticiones al servidor en un promedio de 2.010 segundos, lo que equivale a 2.5 peticiones por segundo. Atendiendo a la cantidad de peticiones por cada segundo que se enviaron, las prestaciones del *hardware* donde se realizaron las pruebas se considera que constituye un resultado satisfactorio.

### 3.3.2 Pruebas funcionales

Se denominan pruebas funcionales 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 [48].

Tabla 7 Caso de prueba Añadir contenido medio AFT

<b>Caso de prueba</b> Añadir contenido medio AFT	
<b>Código de caso de prueba</b>	<b>Nombre de historia de usuario:</b> Añadir contenido medio AFT
<b>Nombre de la persona que realiza la prueba:</b> Hector Daniel Rodríguez Lorenzo	
<b>Descripción de la prueba:</b> Prueba a la funcionalidad añadir contenido medio AFT.	
<b>Entrada/Pasos de ejecución:</b> Se introducen los siguientes datos para crear el medio AFT.	
<p><b>Medio:</b> Teclado</p> <p><b>Identificador:</b> NS-1234567</p> <p><b>Procedencia:</b> Almacén</p> <p><b>Estado:</b> Disponible</p> <p><b>Descripción:</b> Teclado inteligente, color negro</p> <p>El usuario autenticado presiona el botón guardar y si los datos están correctos se guarda medio AFT que podrá ser visualizado desde la vista Listado de medios AFT, si existe algún dato incorrecto el sistema mostrará un mensaje de error y señalará el campo erróneo en rojo para su posterior corrección.</p>	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Tabla 8 Caso de prueba Editar contenido medio útil

Caso de prueba Editar contenido medio útil	
<b>Código de caso de prueba</b>	<b>Nombre de historia de usuario:</b> Editar contenido medio útil
<b>Nombre de la persona que realiza la prueba:</b> Hector Daniel Rodríguez Lorenzo	
<b>Descripción de la prueba:</b> Prueba a la funcionalidad editar contenido medio útil.	
<b>Entrada/Pasos de ejecución:</b> Se modifican los siguientes datos para editar un contenido de tipo medio útil	
<p><b><u>Datos actuales</u></b></p> <p><b>Medio:</b> Vasos</p> <p><b>Cantidad:</b> r12</p> <p><b>Procedencia:</b> Almacén</p> <p><b>Descripción:</b> Vasos plásticos naranjas</p> <p><b><u>Datos después de editados</u></b></p> <p><b>Medio:</b> Vasos</p> <p><b>Cantidad:</b> 150</p> <p><b>Procedencia:</b> Almacén</p> <p><b>Descripción:</b> Vasos plásticos naranjas</p> <p>El usuario autenticado presiona el botón guardar y si los datos están correctos actualiza el medio útil. que podrá ser visualizado desde la vista Listado de medios útil, si existe algún dato incorrecto el sistema mostrará un mensaje de error y señalará el campo erróneo en rojo para su posterior corrección.</p>	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Tabla 9 Caso de prueba Asignar medio útil

Caso de prueba Asignar medio útil	
<b>Código de caso de prueba</b>	<b>Nombre de historia de usuario:</b>

	Asignar medio útil.
<b>Nombre de la persona que realiza la prueba:</b> Hector Daniel Rodríguez Lorenzo	
<b>Descripción de la prueba:</b> Prueba a la funcionalidad Asignar medio útil.	
<b>Entrada/Pasos de ejecución:</b> Desde la vista Listado de útiles.	
El usuario autenticado escoge un medio útil del listado y selecciona la opción Editar.	
<b>Medio:</b> Manteles	
<b>Cantidad:</b> 200	
<b>Procedencia:</b> CC2	
<b>Descripción:</b> Manteles verdes	
Una vez en la vista de edición del medio, se completa el formulario del <i>field collection</i> Asignación, llenando los campos:	
<b>Local:</b> CC1 Comedor 2	
<b>Cantidad prestada:</b> 50	
<b>Recibe:</b> Wendy Díaz Hernández	
Si se desea realizar asignaciones del mismo medio útil a diferentes locales se hace click en la opción Añadir otro elemento, apareciendo los campos que posteriormente serán llenados.	
<b>Local:</b> CC1 Comedor 3	
<b>Cantidad prestada:</b> 60	
<b>Recibe:</b> Jorge Eduardo Pérez Batista	
El usuario autenticado da click en la opción Guardar y si la suma de las cantidades prestadas es menor o igual a la cantidad total del medio se realiza satisfactoriamente la asignación, en caso contrario el sistema lanza un mensaje de error.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

### Resultado de las pruebas funcionales

Para validar que el sistema cumpla con las funciones específicas para las cuales ha sido creado se realizaron cuatro iteraciones donde se encuentran un total de 31 no conformidades, 20 en la primera iteración, 9 en la segunda iteración, 2 en la tercera y ninguna en la cuarta iteración. Al final de

las iteraciones quedan resueltas todas las no conformidades. En la figura 29 se muestran los resultados obtenidos en cada una de las iteraciones de pruebas realizadas al Sistema para la gestión de medios de la sede cubana del concurso regional caribeño del ACM-ICPC.



Figura 29 Comportamiento de las no conformidades por iteración

**Entre las no conformidades detectadas en el proceso de pruebas funcionales se encuentran:**

- Errores de estructuración de los contenidos mostrados en las vistas.
- Los datos introducidos por el usuario de forma incorrecta son guardados en la base de datos sin validación previa.
- Opciones que no funcionan.
- El sistema muestra mensajes de error con datos sobre las variables.
- Los mensajes presentan problemas de idioma.

**3.3.3 Pruebas de seguridad**

Intenta verificar que los mecanismos de protección incorporados en el sistema lo protegerán de hechos de acceso impropio. Durante las pruebas de seguridad, el responsable de la prueba desempeña el papel de un individuo que desea entrar en el sistema. Debe intentar conseguir las claves de acceso por cualquier medio, debe bloquear el sistema, negando así el servicio a otras personas [46].

**Resultados de las pruebas de seguridad**

Con el objetivo de evaluar la seguridad de la solución propuesta se emplea la herramienta Acunetix WVS v8 la cual arrojó los siguientes resultados.

Tabla 10 Resultados de las pruebas de seguridad

Categorías de vulnerabilidades	Cantidad de errores
Formularios HTML sin protección CSRF <sup>13</sup>	29
Credenciales de usuarios enviadas en texto plano	5
Vínculos rotos	47
Campos de contraseña con auto completamiento activado	2
Ajuste de funcionalidades	21
Total	104

La prueba realizada mediante la herramienta Acunetix WVS v8 de nivel 2, arrojó que en la primera iteración se detectan 104 no conformidades, de ellas 52 alertas de riesgo alto, 5 alertas de riesgo medio y 47 de riesgo bajo. 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. Para proporcionar una mayor seguridad se configuró el servidor y se empleó la API de Drupal para la implementación de funciones.

### 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 [37]. El desarrollo de la prueba evidenció la correcta integración de la propuesta de solución con el servidor de correos.

<sup>13</sup> Del inglés: **Cross-site request forgery** . Falsificación de petición en sitios cruzados.

Tabla 11 Resultados de las pruebas de integración

Sistema	Funcionalidades	Resultado de la prueba
Servidor de correo	La integración con el servidor de correo permite el envío de notificaciones a los responsables de los locales.	El Sistema para la gestión de medios de la sede cubana del concurso regional caribeño del ACM-ICPC se integra satisfactoriamente con el servidor de correo, logrando así la comunicación de la administración con los responsables.

### 3.4 Validación de la hipótesis de la investigación

Para la validación de la hipótesis de investigación se utiliza el método criterio de expertos en su variante Delphi [49] empleando los siguientes pasos:

- Identificación de los posibles expertos.
- Selección de los expertos.
- Realización de la consulta a los expertos y procesamiento y valoración de la información obtenida.

Para identificar los posibles expertos se tuvo en cuenta, la experiencia profesional en relación con el desarrollo y trabajo con portales web desarrollados en Drupal 7 y miembros del *staff* de la ACM-ICPC. En la siguiente tabla se muestran los expertos seleccionados.

Tabla 12 Expertos utilizados en la validación de la propuesta de solución

No.	Experto	Entidad	Años de experiencia
1	Ramón Morales Alvares	CIDI	3
2	Nodelvis Hernández Rodríguez	CIDI	3
3	Liliana Martínez Sanabria	CIDI	4
4	Yurelkys de los Ángeles	CIDI	3
5	Dayamí Palma Osoria	CIDI	4
6	José Antonio Castaño Guevara	Dirección de Gestión Tecnológica	5
7	Yuneldis Reyes Velázquez	CIDI	5

Luego de seleccionados los expertos, se sometió a su consideración un instrumento para la validación del Sistema para la gestión de medios de la sede cubana del concurso regional caribeño del ACM-ICPC. El instrumento se compone de 5 sentencias, así como 5 categorías evaluativas que permitan conocer la opinión de los expertos. Las categorías evaluativas empleadas fueron: muy adecuado (MA), bastante adecuado (BA), adecuado (A), poco adecuado (PA) e inadecuado (I).

Se calcula el coeficiente de Kendall que permite analizar la concordancia en las valoraciones realizadas por los expertos [50]. En este caso el coeficiente de concordancia (W) será un índice de la divergencia del acuerdo efectivo entre los expertos. El coeficiente de concordancia de Kendall se obtiene de la expresión:

$$W = 12S / K^2(N^3 - N).$$

$$W = 0.95$$

Donde S representa el cuadrado de las desviaciones medias, K el número de expertos y N el número total de aspectos a evaluar. El valor de W oscila entre 0 y 1. El valor de 1 significa una concordancia de acuerdos total y el valor de 0 un desacuerdo total.

Se aplica además la Prueba de Significación de Hipótesis para comprobar el grado de significación de Kendall, planteándose la hipótesis nula y la alternativa de la siguiente forma: donde H0: no existe concordancia entre los expertos y H1: existe concordancia entre los expertos.

$$x^2 = K(N - 1)W$$

$$x^2 = 26.6$$

El  $x^2$  calculado se compara con el tabulado en la tabla del percentil de la distribución  $x^2$ . Para tener un 95% de confianza se utilizará  $\alpha=0.05$ . Si se cumple que  $x^2_{calculada} < x^2(\alpha, N - 1)$ , entonces se infiere que existe concordancia de criterios entre los expertos al considerar válida la hipótesis alternativa H1. Como se obtiene que  $26.6 < 94.88$  entonces es válida la hipótesis alternativa H1.

Los criterios aportados por los expertos se someten a una prueba estadística no paramétrica que permite concluir qué valoración final tiene cada uno de los aspectos a evaluar. Para los datos anteriores se debe confeccionar una distribución de frecuencia a partir de los datos primarios para cada uno de los aspectos sometidos a consulta [51].

Tabla 13 Distribución de frecuencia para los datos primarios obtenidos

<b>Categorías evaluativas</b>	<b>Frecuencia Absoluta</b>	<b>Frecuencia Relativa</b>
Muy Adecuado	28	0,80
Bastante Adecuado	5	0,14
Adecuado	2	0,06
Poco adecuado	0	0
Inadecuado	0	0

Los resultados obtenidos en la validación de la hipótesis de investigación por criterio de expertos pueden observarse en la Figura 30.



Figura 30 Comportamiento de la valoración de los expertos por categorías evaluadas

De acuerdo con los resultados el 80% de los aspectos analizados fueron valorados de muy adecuado, el 14% de bastante adecuado y un 6% de adecuado. El análisis de los resultados obtenidos de la consulta de expertos permitió identificar criterios muy positivos en las valoraciones realizadas al Sistema para la gestión de medios de la sede cubana del concurso regional caribeño del ACM-ICPC. Al ser evaluados todos los indicadores satisfactoriamente se evidencia la calidad de la solución propuesta.

### 3.5 Conclusiones parciales

Como parte del desarrollo del presente capítulo se determinan las siguientes conclusiones parciales:

- La confección del diagrama de componentes permitió observar la integración de los componentes de *software*.
- Aplicar los estándares de codificación permitió obtener en el sistema un código legible, estándar y fácil de comprender lo que asegura la calidad y facilita un futuro mantenimiento.
- El proceso de validación de la solución propuesta a través de las pruebas de carga y estrés, funcionalidad, seguridad, integración y la consulta a expertos permitió identificar y corregir las no conformidades detectadas para obtener un producto de mayor calidad.

## CONCLUSIONES

- El estudio de los referentes teóricos y el análisis de las diferentes herramientas y tendencias para la gestión de recursos permitió determinar la no existencia de un sistema informático que responda a las necesidades requeridas por el cliente.
- El diseño de la propuesta de solución permitió generar los artefactos más significativos de acuerdo con la metodología de desarrollo de *software* AUP-UCI tomándose como referencia los requisitos detectados.
- La implementación del sistema a través de las herramientas y lenguajes seleccionados permitió obtener una aplicación web capaz de manejar datos referentes a los medios y el despliegue de los mismos durante la competición.
- Las técnicas de validación aplicadas a la propuesta de solución permitieron la detección y corrección de las no conformidades detectadas y evidenciaron que el sistema constituye una solución funcional.
- La validación de la hipótesis con criterio de expertos demostró que el Sistema para la gestión de medios en la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC contribuye a informatizar el control y seguimiento de los medios.

## **RECOMENDACIONES**

Una vez concluida la investigación y el desarrollo de la propuesta de solución, el autor del presente trabajo recomienda:

- Utilizar la solución propuesta para gestionar los medios en futuras ediciones del Concurso Regional Caribeño del ACM-ICPC.
- Agregar funcionalidades que permitan generar reportes por locales y ediciones.
- Implementar una aplicación Android que permita la gestión de los medios en la sede cubana del concurso regional caribeño del ACM-ICPC

## Referencias bibliográficas

1. HERNÁNDEZ MOYA, A. L. Sistema de gestión de procesos organizativos de la Sede Cubana del Concurso Regional Caribeño del ACM-ICPC. Trabajo de Diploma. Universidad de las Ciencias Informáticas. La Habana, 2016.
2. Baylor. World Finals Champions. 2016. [En línea] [Consultado el: 27 de noviembre de 2016] Disponible en: <https://icpc.baylor.edu/community/world-finals-champions>.
3. RIPOLL MÉNDEZ, DOVIER A. Inauguración del Concurso Regional Caribeño. 2009, La Habana.
4. Cubadebate. Realizarán en la Universidad de Ciencias Informáticas concurso internacional de Programación. 2009. [En línea] [Consultado el: 27 de noviembre de 2016] Disponible en: <http://www.cubadebate.cu/noticias/2009/10/19/realizaran-universidad-ciencias-informaticasconcurso-internacional-programacion-acm-icpc/>
5. RIPOLL MÉNDEZ, D. A. Participación del Caribe en el ACM-ICPC a partir del año 2009. [En línea] [Consultado el: 27 de noviembre de 2016] Disponible en: <http://coj.uci.cu/downloads/acm-icpc-carib/ParticipCaribACM-ICPC.pdf>
6. GISBERT, M. D. T. Y. y GROSS, R. G.-P. Y. (1974). *Pequeño Larousse Ilustrado*. [En línea], 2016. [Consultado el: 25 de febrero de 2017]. Disponible en: <https://archive.org/details/PequeoLaroussellustrado>
7. Sistemas de Gestión [En línea], 2016. [Consultado el: 25 de febrero de 2017] Disponible en: <http://www.consultoresdesistemasdegestion.es/sistemas-de-gestion/>
8. Real Academia Española [En línea], 2017. [Consultado el: 25 de febrero de 2017]. Disponible en: <http://dle.rae.es/?id=M2v6jgO>
9. FÁBREGA AVILA, C. Implementación del módulo de inventario del subsistema Activo Fijo Tangible del Sistema Integral de Gestión CEDRUX. Trabajo de Diploma. Universidad de las Ciencias Informáticas. La Habana, 2010.
10. MARTÍNEZ, A. Metodologías para el desarrollo de software. [En línea]. 2016. [Consultado el: 28 de noviembre de 2016] Disponible en: <http://es.slideshare.net/yeltsintorres18/metodologias-para-el-desarrollo-del-software>
11. CADAVID, A. Revisión de metodologías ágiles para el desarrollo de software. *Prospectiva*, 2013, vol. 11, no 2, p. 30-39. [Consultado el: 28 de noviembre de 2016] Disponible en: <http://ojs.uac.edu.co/index.php/prospectiva/article/viewFile/36/21>

12. RODRÍGUEZ SÁNCHEZ, T. Metodología de desarrollo para la Actividad productiva de la UCI v1.2. 2015, La Habana
13. Definición de Herramienta - Significado y definición de Herramienta. [En línea], 2016. [Consultado el: 25 de febrero de 2017]. Disponible en: <https://www.mastermagazine.info/termino/5234.php>
14. GIL RODRÍGUEZ, F. *Experto en Drupal: Nivel Inicial*. Forcontu S.L., 2011. 363 p.
15. Karakana. [En línea], 2016. [Consultado el: 28 de noviembre de 2016]. Disponible en: <http://www.karakana.es/disenio-web/blog/sistemas-de-gestion-de-contenidos-cms-drupal>
16. Drupal. [En línea], 2016. [Consultado el: 20 de noviembre de 2016]. Disponible en: <https://www.drupal.org/drupal-7.0/es>.
17. PANDIYAN, M., SINGHAL, M. Development of a Lightweight Learning Management System using open source software Drupal, 2015.
18. ALVAREZ, S. Sistemas gestores de bases de datos. [En línea] [Consultado el: 23 de noviembre de 2016] Disponible en: <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
19. Página Oficial de Postgres SQL. PostgreSQL. About. [En línea] [Consultado el: 23 de noviembre de 2016] Disponible en: <http://www.postgresql.org/about/>.
20. PHP y MySQL: Tecnología para el desarrollo de aplicaciones web. -Ángel Cobo -Google Libros.[En línea] [Consultado el : 23 de noviembre de 2016] Disponible en: [http://www.google.com/cu/books?hl=es&lr=&id=zMK3GOMOpQ4C&oi=fnd&pg=PR17&dq=php+es+un+lenguaje+de+programaci%C3%B3n+&ots=FeizY3Hfvi&sig=6Fb7-SA1M8ZKqOyahPKJIHRplhk&redir\\_esc=y#v=onepage&q=php%20es%20un%20lenguaje%20de%20programaci%C3%B3n&f=false](http://www.google.com/cu/books?hl=es&lr=&id=zMK3GOMOpQ4C&oi=fnd&pg=PR17&dq=php+es+un+lenguaje+de+programaci%C3%B3n+&ots=FeizY3Hfvi&sig=6Fb7-SA1M8ZKqOyahPKJIHRplhk&redir_esc=y#v=onepage&q=php%20es%20un%20lenguaje%20de%20programaci%C3%B3n&f=false).
21. KEMPPAINEN, T, Data Archive Project [en línea]. 2015. Helsinki: Helsinki Metropolia University of Applied Sciences. [Consultado el: 21 de noviembre de 2016] Disponible en: [http://www.theseus.fi/bitstream/handle/10024/93069/Kemppainen\\_TimoPekka.pdf?sequence=1](http://www.theseus.fi/bitstream/handle/10024/93069/Kemppainen_TimoPekka.pdf?sequence=1).
22. KHOLODKOV, V. Nginx Essentials. Packt Publishing Ltd, 2015. ISBN 978-1-78528-953-8
23. Nginx. [En línea], 2015. [Consulta: 21 noviembre 2016]. Disponible en: <https://www.nginx.com>
24. NEDELCO, C. Nginx HTTP Server. 2010. Packt Publishing. ISBN 978-1-84951-086-8.
25. Cherokee Web Server | Why Cherokee? | Cherokee Documentation [En línea], 2013. [Consultado el: 22 de noviembre de 2016]. Disponible en: [http://cherokee-project.com/doc/basics\\_why\\_cherokee.html](http://cherokee-project.com/doc/basics_why_cherokee.html).

26. URIBE, C.A. Balanceo de Carga de Aplicaciones Web con Apache Web Server [en línea]. 2014. [Consultado el: 21 de noviembre de 2016]. Disponible en: <http://repositorio.uniquindio.edu.co/bitstream/123456789/486/5/Proyecto%20547%20Informe%20Final.pdf>
27. Visual Paradigm. [En línea], 2015. [Consultado el: 21 de noviembre de 2016]. Disponible en: <http://www.visual-paradigm.com>
28. HEFFELFINGER, D. Java EE 7 Development with NetBeans 8. Packt Publishing Ltd, 2015. [Consultado el: 21 de noviembre de 2016]. Disponible en: [https://books.google.com.cu/books?hl=es&lr=&id=VY92BgAAQBAJ&oi=fnd&pg=PP1&dq=Java+EE+7+Development+with+NetBeans+8+&ots=P4THdparyA&sig=EE\\_ZjEVFhxNIFqv7KbxRmMjKzk&redir\\_esc=y#v=onepage&q=Java%20EE%207%20Development%20with%20NetBeans%208&f=false](https://books.google.com.cu/books?hl=es&lr=&id=VY92BgAAQBAJ&oi=fnd&pg=PP1&dq=Java+EE+7+Development+with+NetBeans+8+&ots=P4THdparyA&sig=EE_ZjEVFhxNIFqv7KbxRmMjKzk&redir_esc=y#v=onepage&q=Java%20EE%207%20Development%20with%20NetBeans%208&f=false)
29. PRESSMAN, R. 2002. Ingeniería del software, un enfoque práctico. S.l.: s.n.
30. DSA. [En línea], 2016. [Consultado el: 2 de abril noviembre de 2016]. Disponible en: <http://dsav.net/wp-content/uploads/2013/02/Acunetix-WVS-castellano.pdf>
31. ApacheJMeter. [En línea], 2016. [Consulta: 26 noviembre 2016]. Disponible en: <http://jmeter.apache.org>
32. World Wide Web Consortium (W3C). What is HTML? [En línea]. [Consultado el: 22 de noviembre de 2016]. Disponible en: <http://www.w3.org/html/>.
33. World Wide Web Consortium (W3C). What is CSS? [En línea]. [Consultado el: 22 de noviembre de 2016]. Disponible en: <http://www.w3.org/Style/CSS/>.
34. Colectivo de Autores. Manual de PHP. The PHP Documentation [En línea]. [Consultado el: 22 de noviembre de 2016]. Disponible en: <http://www.php.net/manual/es/index.php>.
35. Object Management Group. Welcome To UML Web Site!.2016. [En línea] [Consultado el: 22 de noviembre de 2016]. Disponible en: <http://www.uml.org/>.
36. LARMAN, C. UML y Patrones. 2003, vol. 2. [En línea]. [Consultado el: 17 de febrero de 2017]. Disponible en: <<http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDiseno.pdf>>.
37. SOMMERVILLE, I. 2005. Ingeniería del software. Séptima edición. S.l.: PEARSON EDUCACIÓN, S.A., Madrid, 2005, ISBN 84-7829-074-5.

38. Glinz, M.: *On-Non-Functional Requirements*. 15th IEEE International Requirements Engineering Conference. 21 -26. DOI 10.1109/RE.2007.45. IEEE Computer Society, (2007).
39. Suaza, K. V. Definición de equivalencias entre historias de usuario y especificaciones en UNLENCEP para el desarrollo ágil de software [Tesis]. Universidad Nacional de Colombia: Medellín, Colombia, 2013.
40. VANDYK, J. An Introduction to Drupal Architecture. In DrupalCamp Des Moines. Iowa, 2011.
41. Blázquez, M. Modelo entidad-relación ER. [artículo]. Fundamentos y Diseño de Bases de Datos. [En línea]. [Consultado el: 17 de febrero de 2017]. Disponible en: <http://ccdoc-basesdedatos.blogspot.com/2013/02/modelo-entidad-relacion-er.html>
42. JACOBSON, J. 2000. El proceso unificado de desarrollo de Software. Madrid: PEARSON EDUCACIÓN,S.A., 2000.
43. RIVERA ALVA, E. 2008. Arquitectura de Software II. Diagramas de Componentes y Despliegue. [En línea]. [Consultado el: 17 de febrero de 2017]. Disponible en: <http://es.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue>.
44. Pruebas de software. 2005. [En línea] [Consultado el: 2 de abril 2017]. Disponible en: <http://pruebasdesoftware.com/laspruebasdesoftware.htm>.
45. Guía digital. 2008. [En línea] [Consultado el: 2 de abril 2017]. Disponible en: <http://www.guiadigital.gob.cl/articulo/pruebas-de-carga>
46. PRESSMAN, R. 2005. Software Engineering A Practitioner's Approach Seventh Edition. [En línea] [Consultado el: 2 de abril 2017]. Disponible en: <http://www.vumultan.com/Books/CS605-Software%20Engineering%20Practitioner%E2%80%99s%20Approach%20%20by%20Roger%20S.%20Pressman%20.pdf>
47. SEGURA, N. Significado de las distribuciones muestrales en textos universitarios de estadística. Revista electrónica de investigación en educación en ciencias, 2012, vol. 7, no 2, p. 54-71.
48. Pruebas Funcionales.2009. [En línea] [Consultado el: 2 de abril 2017]. Disponible en: [http://www.calidadyssoftware.com/testing/pruebas\\_funcionales.php](http://www.calidadyssoftware.com/testing/pruebas_funcionales.php).
49. Sánchez, S. Estrategia de soporte técnico para el proceso de migración a código abierto en los Organismos de la Administración Central del Estado. 2015. Maestría. S.l.: Universidad de las

Ciencias Informáticas.

50. SAMPIERI, R, et al. Metodología de la investigación. McGraw-Hill Education, 2014.

51. CASTRO, L. Guía de gestión del riesgo tecnológico para el tratamiento de la seguridad durante el proceso de desarrollo de software. 2014. [Consultado el:2 de abril de 2017].

## Bibliografía

- ACEBAL ALARCÓN, J. E. Informatización del proceso Vales de devolución del subsistema Inventario-Cedrux. Trabajo de Diploma. Universidad de las Ciencias Informáticas. La Habana, 2013.
- Arquitectura | Tutorial Drupal [En línea] [Consultado el: 22 de noviembre de 2016]. Disponible en: <http://www.cursosdrupal.com/content/arquitectura>
- Asset Tracking Software - EZOfficeInventory. 2016. [En línea] [Consultado el: 2 de diciembre de 2016] Disponible en: <http://www.ezofficeinventory.com/>
- CHÁVEZ MARRERO, Y. Módulo de publicación automática de contenidos web en redes sociales para Drupal 7. Trabajo de Diploma. Universidad de las Ciencias Informáticas. La Habana, 2016.
- Cubadebate. ACM-ICPC: La competencia de programadores donde Cuba lideró a Latinoamérica .2016. [En línea] [Consultado el: 2 de diciembre de 2016]. Disponible en: <http://www.cubadebate.cu/noticias/2016/11/16/acm-icpc-la-competencia-de-programadores-donde-cuba-lidero-a-toda-latinoamerica-fotos-y-videos/>.
- Cubadebate. Competencia Internacional Universitaria de Programación (ACM-ICPC). 2016. [En línea] [Consultado el: 2 de diciembre de 2016] Disponible en: <http://www.cubadebate.cu/etiqueta/competencia-internacional-universitaria-de-programacion-acm-icpc/>.
- Cubadebate. Equipo de Universidad de La Habana hace historia en torneo de programación caribeño. 2016. [En línea] [Consultado el: 2 de diciembre de 2016] Disponible en: <http://www.cubadebate.cu/noticias/2016/11/13/equipo-de-universidad-de-la-habana-hace-historia-en-torneo-de-programacion-caribeno/>
- Drupal 6/7 programming from an object-oriented perspective [En línea]. [Consultado el: 15 de febrero de 2017]. Disponible en: <https://www.drupal.org/docs/7/creating-custom-modules/drupal-67-programming-from-an-object-oriented-perspective>.
- Drupal Hispano | Comunidad de usuarios de Drupal.2016. [En línea] [Consultado el: 24 de noviembre de 2016] Disponible en: <http://drupal.org/es/>
- Drupal. Mission and principles.2016. [En línea] [Consultado el: 25 de noviembre de 2016] Disponible en: <https://www.drupal.org/about/mission-and-principles>

- FÁBREGA AVILA, C. Implementación del módulo de inventario del subsistema Activo Fijo Tangible del Sistema Integral de Gestión CEDRUX. Trabajo de Diploma. Universidad de las Ciencias Informáticas. La Habana, 2010.
- GIL RODRÍGUEZ, F. *Experto en Drupal: Nivel Inicial*. Forcontu S.L., 2011. 363 p.
- HECHAVARRÍA BREAL, R; FUENTES CASTILLO, Y. Módulo de Facturación Telefónica para la plataforma PLATEL Trabajo de Diploma. Universidad de las Ciencias Informáticas. La Habana, 2014.
- Inventoria Stock Control and Inventory Software Screenshots. 2016. [En línea] [Consultado el: 3 de diciembre de 2016] Disponible en: <http://www.nchsoftware.com/inventory/screenshots.html>
- Inventory Tracker Plus. 2016. [En línea] [Consultado el: 2 de diciembre de 2016] Disponible en: <https://inventory-tracker-plus.en.softonic.com/>
- Juventud Rebelde. Destacan aplicación de software cubano para el control contable.2015. [En línea] [Consultado el: 3 de diciembre de 2016] Disponible en: <http://www.juventudrebelde.cu/ciencia-tecnica/2012-08-20/destacan-aplicacion-de-software-cubano-para-el-control-contable/>.
- Mejora de procesos de software.2008. [En línea] [Consultado el: 6 de diciembre de 2016] Disponible en: <http://mejoras.prod.uci.cu/>.
- PostGreSQL vs. MySQL [En línea] [Consultado el: 25 de febrero de 2017]. Disponible en: <https://danielpecos.com/documents/postgresql-vs-mysql/>
- RODRÍGUEZ CALERO, O. Comparativa entre gestores de contenidos (CMS). 2016. [En línea] [Consultado el: 24 de noviembre de 2016] Disponible en: <http://www.hiberus.com/blog/comparativa-entre-gestores-de-contenidos-cms>.
- SILVENTE LAO, Y. J. Módulo para la detección de contenido duplicado en portales web. Trabajo de Diploma. Universidad de las Ciencias Informáticas. La Habana, 2015.
- XABAL. Excriba Gestor de documentos administrativos.2013. [En línea] [Consultado el: 6 de diciembre de 2016] Disponible en: <https://excriba.prod.uci.cu/page/>.