

**Universidad de las Ciencias Informáticas  
FACULTAD 6**



**Módulo de personalización para la interfaz  
gráfica de la Plataforma VideoWeb.**

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

**Autor:** Tailén Rego Castillo

**Tutor:** Ing. Danis Rego Castillo

**La Habana, Junio 2012**



*“Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido...”*

*“Che”*

## **Declaración de Autoría**

Declaro que Tailén Rego Castillo es la única autora de este trabajo titulado “Módulo de personalización para la interfaz gráfica de la Plataforma VideoWeb.” y autorizo a la Facultad 6 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

---

Tailén Rego Castillo

---

Ing. Danis Rego Castillo

## Agradecimientos

*A mi Suny y a mi Bicho por haberme apoyado siempre en mis decisiones, por haber dejado que cometiera mis propios errores y por todo el amor que me han dado. A ustedes que luchan cada día por mi bienestar, a quienes les debo todo lo que soy, porque ustedes son mi guía y mi ejemplo a seguir en la vida. Los amo con toda mi alma.*

*A mi hermano, por ser el mejor hermano del mundo. Has sido un punto de apoyo muy importante para mí y un maravilloso ejemplo a seguir. Gracias por tus consejos y por siempre tener confianza en mí y darme ánimos para seguir adelante. Aunque casi siempre estamos fajados yo te quiero con todo mi corazón.*

*A mis abuelos por su cariño y constante preocupación y a mi abuelo Pancho que ya no está con nosotros. Siempre estoy pensando en ustedes.*

*A mi tía la Poderosa y su familia por ser tan buenos conmigo y preocuparse tanto por mí.*

*A mis vecinos y demás familiares que siempre estuvieron ahí para mí.*

*A Eny, Emy, Katia, La Chuchi, Mirita, Day, Yany que supieron soportarme y luchar conmigo todo este tiempo que pasamos juntas. Las quiero mucho y siempre las llevaré en mi corazón.*

*A los muchachos del 9109, juntos dimos los primeros pasos en esta universidad, momentos que quedaron grabados en mi mente y a los otros que fui conociendo durante la carrera.*

*A mis amistades de la UCI, los que están y los que ya no están, que me ayudaron mucho no sólo en el desarrollo de mi tesis, sino a lo largo de estos cinco años de estudio.*

*A los profesores del tribunal y en especial a mi tutor por apoyarme y guiarme, gracias por sus consejos y recomendaciones.*

*A la Revolución y al Comandante Fidel Castro, por la maravillosa idea de crear esta escuela.*

*Agradezco a todos aquellas personas que de una forma u otra me han ayudado y me han apoyado durante toda mi carrera, a toda mi familia, mis amigos y profesores. A todos muchas gracias.*

**Dedicatoria**

*A mis padres,  
Por la educación que me han dado, la confianza que siempre han depositado en mí y por ser las mejores personas del mundo. Por su apoyo y por ese amor tan grande que me han brindado siempre, los quiero.*

*A mi hermano,  
Porque es la fuerza que me inspira a ser cada día mejor, por ayudarme y apoyarme, y por creer siempre en mí y no abandonarme ni un momento.*

*A mis abuelos,  
Por haberme dado la posibilidad de estar hoy frente a ustedes y por ser las personas más especiales que han pasado por mi vida.*

*A mis amigas,  
Gracias por estar siempre presentes cuando las necesité.*

*A ti,  
Que lo estas leyendo hoy.*

*A mí,  
Que gracias a la gran investigación que realicé me permitió poder redactar este trabajo.*

## Resumen

La personalización de aplicaciones *web* consiste en desarrollar sitios *web* cuya interfaz y funcionalidades se adapten y respondan a las exigencias y necesidades de los usuarios. En Cuba existen muchas aplicaciones que requieren ser personalizadas dentro de las cuales podemos encontrar a la Plataforma VideoWeb, que es un producto destinado a la gestión y transmisión de contenidos audiovisuales a través de la red de datos utilizando tecnología *streaming*, pensada para universidades y otras instituciones que deseen hacer llegar materiales audiovisuales a los usuarios de la red. La Plataforma VideoWeb necesita una herramienta que le permita a los administradores del sistema poder cambiar la apariencia de la interfaz de la plataforma de manera sistemática, para lo cual se decidió implementar un módulo que permita optimizar el proceso de personalización al que son sometidas las plantillas que tiene definida la misma y que se utilizan para presentar a los usuarios las diferentes ofertas que tiene la Plataforma. El producto final permite a los administradores de la plataforma poder modificar el banner, reemplazar el logo, cambiar la tipografía en cuanto al tipo de letra, tamaño y color de la misma, así como los subtítulos de la página y además el color de fondo de la plantilla y del banner por una imagen y viceversa.

**Palabras claves:** *Marketing*, Personalización, Plataforma.

**Índice de Tablas**

<i>Tabla 1: Descripción del CU Cambiar Fondo de Página.....</i>	<i>28</i>
<i>Tabla 2: Descripción del CU Modificar Banner.....</i>	<i>30</i>
<i>Tabla 3: Descripción del CU Configurar Menú Principal.....</i>	<i>31</i>

**Índice de Figuras**

<i>Figura 1: Diagrama de Clases del Dominio.....</i>	<i>22</i>
<i>Figura 2: Diagrama de Casos de Uso del Sistema.....</i>	<i>25</i>
<i>Figura 3: Vista del Patrón Arquitectónico Modelo en dos capas.....</i>	<i>33</i>
<i>Figura 4: Diagrama de clases del análisis del CU Cambiar Fondo de Página.....</i>	<i>35</i>
<i>Figura 5: Diagrama de clases del análisis del CU Modificar Banner.....</i>	<i>35</i>
<i>Figura 6: Diagrama de clases del análisis del CU Configurar Menú Principal.....</i>	<i>35</i>
<i>Figura 7: Diagrama de clases del diseño.....</i>	<i>36</i>
<i>Figura 8: Diagrama de Secuencia del CU Modificar Fondo de Página.....</i>	<i>38</i>
<i>Figura 9: Diagrama de Secuencia del CU Modificar Banner.....</i>	<i>38</i>
<i>Figura 10: Diagrama de Secuencia del CU Modificar Menú Principal.....</i>	<i>38</i>
<i>Figura 11: Diagrama de Despliegue.....</i>	<i>39</i>
<i>Figura 12: Diagrama de Componentes.....</i>	<i>40</i>

**Índice de Contenidos**

<b>Introducción</b> .....	<b>1</b>
<b>Capítulo I: Fundamentación Teórica</b> .....	<b>7</b>
1.1. <i>Introducción</i> .....	7
1.2. <i>La personalización de aplicaciones web en el mundo</i> .....	7
1.3. <i>Conceptos Asociados al Dominio del Problema</i> .....	8
1.3.1. <i>VideoWeb</i> .....	9
1.3.2. <i>Aplicación Informática</i> .....	9
1.3.3. <i>Web</i> .....	9
1.3.4. <i>Personalización</i> .....	9
1.4. <i>Soluciones o sistemas similares a nivel mundial</i> .....	10
1.4.1. <i>Artisteer</i> .....	10
1.4.2. <i>Microsoft Expression</i> .....	11
1.4.3. <i>Web Easy</i> .....	11
1.5. <i>Metodología de Desarrollo del Software</i> .....	12
1.5.1. <i>Proceso Unificado de Desarrollo (RUP)</i> .....	13
1.6. <i>Lenguaje Unificado de Modelado (UML)</i> .....	14
1.7. <i>Herramientas CASE</i> .....	14
1.8. <i>Lenguajes de Programación</i> .....	16
1.9. <i>Entorno de Desarrollo Integrado (IDE)</i> .....	17
1.9.1. <i>NetBeans 6.9</i> .....	17
1.10. <i>Tecnologías para el Desarrollo de Aplicaciones</i> .....	18
1.10.1. <i>CMS Drupal 6.14</i> .....	18
1.11. <i>Servidor Web Apache 2.2</i> .....	19
1.12. <i>Conclusiones</i> .....	19
<b>Capítulo II: Caracterización del Sistema Propuesto</b> .....	<b>21</b>
2.1. <i>Introducción</i> .....	21
2.2. <i>Modelo de Dominio</i> .....	21
2.3. <i>Breve descripción del Diagrama de Clases del Dominio</i> .....	22
2.4. <i>Especificación de los Requisitos del Sistema</i> .....	23
2.4.1. <i>Requisitos Funcionales</i> .....	23



2.3.2. Requisitos No Funcionales .....	24
2.5. Diagrama de Casos de Uso del Sistema.....	25
2.5.1. Descripción de los Casos de Uso del Sistema.....	26
<b>Capítulo III: Análisis y Diseño del Sistema .....</b>	<b>32</b>
3.1. Introducción .....	32
3.2. Patrón Arquitectónico.....	32
3.3. Patrones de Diseño .....	33
3.4. Diagramas de Clases del Análisis.....	34
3.5. Diagrama de Clases del Diseño.....	35
3.6. Diagramas de Interacción .....	37
3.6.1. Diagramas de Colaboración.....	37
3.6.2. Diagramas de Secuencia.....	37
3.7. Diagrama de Despliegue .....	38
3.8. Conclusiones .....	39
<b>Capítulo IV: Implementación y Pruebas .....</b>	<b>40</b>
4.1. Introducción .....	40
4.2. Diagrama de Componentes .....	40
4.3. Estándares de Codificación .....	41
Criterios de Calidad.....	41
Otros Comentarios Internos: .....	42
Aspectos específicos sobre PHP .....	42
4.4. Pruebas de Software .....	44
4.4.1. Pruebas de Caja Negra .....	45
4.5. Conclusiones .....	50
<b>Conclusiones Generales .....</b>	<b>51</b>
<b>Recomendaciones .....</b>	<b>53</b>
<b>Referencias Bibliográficas .....</b>	<b>54</b>
<b>Bibliografía.....</b>	<b>57</b>
<b>Anexos.....</b>	<b>60</b>
<b>Glosario de Términos .....</b>	<b>67</b>

## **Introducción**

La informática es la disciplina que estudia el tratamiento automático de la información utilizando dispositivos electrónicos y sistemas computacionales, se aplica a muchas áreas como son la gestión de negocios, el almacenamiento y consulta de información, monitorización y control de procesos, robots industriales, comunicaciones, control de transportes, investigación, desarrollo de juegos, diseño computarizado y herramientas multimedia. (RAE)

En Cuba, la informática comenzó su desarrollo a finales de la década del 50 con los programas de formación del personal informático, la fabricación de la primera computadora cubana y el Programa de Informatización de la Sociedad. En sus inicios los sistemas con los que se trabajaba tenían poco desarrollo y el personal con que se contaba resultaba de muy baja calificación, pero no fue hasta el año 1968 que empezaron a materializarse decisiones importantes en este campo. En los últimos tiempos, la informática se ha convertido en el motor impulsor del proceso de globalización en el cual está inmerso todo el mundo, por lo que se hace necesario preparar a las nuevas generaciones para la asimilación y utilización de dicha tecnología en proyectos investigativos que ayuden al desarrollo de la economía del país.

Con el transcurso de los años el desarrollo de la informática en el país se ha incrementado, ejemplo de esto fue la creación de los Joven Club de Computación y Electrónica, iniciativa del Comandante en Jefe Fidel Castro Ruz el 8 de septiembre de 1987 y considerada la niña de los ojos de la Revolución en la computación. Más tarde, en el 2002 llegó la Universidad de las Ciencias Informáticas (UCI), con un novedoso modelo de formación que combina el estudio con la producción y la investigación, donde se desarrollan productos para la exportación que ayuden a elevar la economía del país y productos nacionales destinados a la sustitución de importaciones y elevar la informatización de la sociedad cubana.

Muchos son los programas que brinda esta disciplina con los cuales las personas pueden realizar diversos trabajos y beneficiarse con el resultado de los mismos. Una parte muy importante de esta ciencia es la personalización de aplicaciones *web*, proceso que ha tenido gran auge y desarrollo mundialmente. Las aplicaciones *web* son desarrolladas para vender y transmitir información de todo tipo a todos los usuarios, son útiles y beneficiosas pero también pueden ser usadas en contra de personas y entidades importantes.

---

En la actualidad, la personalización de aplicaciones *web* constituye un factor importante y prácticamente obligatorio en las aplicaciones *web* de hoy día. La personalización comprende todo un sinnúmero de aspectos y agrega una nueva dimensión a los diseños de las aplicaciones *web*. Este es un proceso que ayuda a construir una nueva imagen de las organizaciones en el mundo de la economía y con esto aumenta la reputación de las mismas en el mercado mundial.

La personalización de aplicaciones *web* es desde hace varias décadas uno de los pilares del marketing digital que permite desarrollar sitios *web* cuya interfaz y funcionalidades se adapten a los usuarios en función de sus expectativas, necesidades, preferencias y deseos. La personalización conlleva tanto el mantenimiento de la información contextual, formada por las circunstancias de acceso y el perfil del usuario entre otros factores, como el mecanismo de adaptación del sistema al estado cambiante del contexto. La personalización supone, por tanto, un factor de complejidad añadido al desarrollo de sistemas *web*, complicando y encareciendo su desarrollo, mantenimiento y evolución. (Bru, 2012)

La personalización se concibe como una nueva dimensión en el desarrollo de sistemas *web* que afecta a todos sus niveles, aumentando la complejidad del sistema en gran medida. Los sistemas *web* personalizables, por tanto, se hacen más difíciles de desarrollar, mantener y modificar. Las ventajas que brinda este proceso permite que los usuarios sientan que una parte de la *web* les pertenece, hace la *web* más interesante pues los usuarios la configuran a su gusto, además le da frescura e importancia a la *web* pues las modificaciones de los usuarios permiten que los contenidos se actualicen con mayor frecuencia. (Sacristán, 2006)

La interfaz gráfica de las aplicaciones *web* ayuda a los usuarios a identificar donde están y que deben hacer. La personalización va más allá de la estética que tenga la aplicación. Los íconos demasiado pequeños, los menús ilegibles o los colores que se confunden reducen la navegación y disminuyen el rendimiento de los usuarios para con la aplicación. Mejorar la visión de los elementos de la pantalla resulta fundamental y no solamente para aquellas personas que tienen limitaciones visuales sino para todos en general, de esta manera los usuarios se sienten más comprometidos con la aplicación.

El diseño y presentación de la interfaz gráfica en las aplicaciones *web*, juegan un rol considerablemente significativo a la hora de crear las aplicaciones usables y amigables. Para realizar la personalización de la interfaz de las aplicaciones se debe trabajar sobre dos componentes principales del diseño de la interfaz de las herramientas *web*: el tema y las imágenes. En el proceso de

personalización, las imágenes se usan para mostrar logos, botones e íconos. Hoy día se pueden encontrar cientos de aplicaciones que reclaman ser totalmente personalizadas a diferentes perfiles de usuarios o individuos.

Existen modelos de personalización que propician que el proceso de personalización tenga resultados satisfactorios y relevantes. La personalización de la interfaz es uno de los modelos de personalización mediante el cual se les da la posibilidad a los usuarios de que puedan personalizar los elementos que componen la interfaz del usuario. Es importante aclarar que no todos los usuarios tienen los mismos gustos ni están interesados de la misma forma por los contenidos de la interfaz, por lo que sería muy importante permitir que los usuarios aporten sus ideas para una redefinición de la misma y de esta manera conseguir repetidas visitas.

Otro de los modelos es la personalización de contenidos mediante la cual se les muestra a los usuarios la información adecuada de acuerdo a alguna preferencia, hábito o gusto que haya sido recogido en alguna encuesta o mediante otra vía de manera que se pueda mejorar la atención y el trato a los clientes. (IWorld.com, 2006)

Actualmente existen entidades y usuarios independientes que se dedican al proceso de la personalización de aplicaciones *web*. La Universidad de las Ciencias Informáticas es la primera universidad creada al calor de la Batalla de Ideas. Es un centro educativo y productivo que surgió con el objetivo de brindar servicios y desarrollar soluciones de *software* que permitan automatizar procesos y proporcionar avances en la economía del país. La universidad está conformada por diez facultades, siete en la sede principal y tres regionales en las provincias de Artemisa, Ciego de Ávila y Granma.

La Facultad 6 es una de las facultades de la sede principal y está conformada por los centros GEySED y DATEC. El centro de desarrollo GEySED se creó con el objetivo de desarrollar productos y soluciones informáticas en el área del procesamiento de las Señales Digitales y la Geoinformática. El Departamento de Señales Digitales atiende varios proyectos productivos dentro de los cuales se encuentra VideoWeb con el producto "Plataforma VideoWeb".

La Plataforma VideoWeb es un producto destinado a la gestión y transmisión de contenidos audiovisuales a través de la red de datos utilizando tecnología *streaming*, pensada para universidades y otras instituciones que deseen hacer llegar materiales audiovisuales a los usuarios de la red. Fue

desarrollada en su totalidad empleando herramientas y tecnologías libres, por lo que se pretende que en un futuro pueda sustituir a Inter-Nos, sitio de gran relevancia dentro de la UCI como medio de apoyo a la docencia y baluarte en el desarrollo cultural de los estudiantes y trabajadores de la Universidad.

Uno de los problemas actuales que más sufre la plataforma es la poca flexibilidad de los subsistemas *web*, son poco intuitivos en cuanto a los intereses particulares de cada usuario o cliente. Una vez terminado el *software* el cliente no podrá cambiar ni modificar el entorno de los productos que se entregan hasta satisfacer sus gustos y necesidades individuales lo que podría provocar en cierta medida que los usuarios quedaran insatisfechos con el producto entregado y sería fatal para la entidad desarrolladora pues su respeto en el mundo del desarrollo de *software* depende del resultado de sus trabajos.

Para la solución de estos problemas existen sistemas como el Artisteer, Microsoft Expression y *Web Easy*, que son herramientas que permiten crear y personalizar plantillas de interfaces gráficas de alto nivel, ajustando así la aplicación a los intereses del usuario, pero no pueden ser utilizadas debido a que son propietarias y las que se requieren tienen que ser libres y multiplataforma.

Esta necesidad que tiene el proyecto de personalizar y configurar la interfaz gráfica de la plataforma hace que el presente trabajo de diploma busque dar solución al siguiente **Problema a resolver**: ¿Cómo facilitar la personalización de la interfaz gráfica de la Plataforma VideoWeb?

Para el desarrollo de la investigación se define como **objeto de estudio**: los procesos de personalización de las aplicaciones *web* y como **campo de acción**: la automatización de los procesos de personalización de la interfaz gráfica de la plataforma VideoWeb.

El **objetivo general** de este trabajo es implementar un módulo de personalización para la interfaz gráfica de la plataforma VideoWeb que ayude a que el trabajo con la plataforma sea más completo.

Se **defiende la idea** de que con la personalización de la interfaz de la Plataforma VideoWeb aumentará la flexibilidad de la misma pudiéndose ajustar así a los intereses de los usuarios.

Para dar cumplimiento al objetivo general se tienen en cuenta las siguientes **tareas de la investigación**:

---

1. Caracterizar los sistemas de personalización de aplicaciones *web*.
2. Caracterizar la metodología, herramientas y lenguajes de programación a utilizar en la implementación del módulo.
3. Describir los requisitos funcionales y no funcionales.
4. Identificar los subsistemas que soporten las funcionalidades básicas del módulo de personalización.
5. Diseñar las clases a implementar del módulo de personalización de la interfaz gráfica.
6. Implementar el módulo de personalización de la interfaz gráfica.
7. Planificar las pruebas al módulo de personalización.
8. Realizar las pruebas al módulo de personalización.
9. Evaluar los resultados obtenidos de las pruebas.

Para dar cumplimiento a las tareas de la investigación y con el objetivo de obtener conocimientos necesarios y profundos que hagan posible la materialización del objetivo general, se han empleado diferentes métodos científicos de investigación:

**Métodos Teóricos:**

**Histórico-Lógico:** Método utilizado con el objetivo de caracterizar las soluciones existentes sobre los sistemas de personalización de aplicaciones *web*.

**Analítico-Sintético:** Método utilizado con el objetivo de analizar, comprender y sintetizar la información y documentación relacionada con el tema de investigación, permitiendo extraer los elementos necesarios y más importantes relacionados con el objeto de estudio.

**Modelación:** Método aplicado en el desarrollo de la investigación para describir los distintos procesos, mediante la creación de los modelos elaborados en el proceso de desarrollo del *software* que ofrecen la posibilidad de crear abstracciones para explicar la realidad. Se hace visible en el trabajo al crear modelos como: Casos de Uso del Sistema, Dominio, Análisis y Diseño.

**Métodos Empíricos:**

**Entrevista:** Método aplicado durante el desarrollo de la investigación a profesores y estudiantes

---

pertenecientes al proyecto con el objetivo de obtener información acerca del tema de investigación que se propone. (Ver Anexo 4)

**El trabajo está conformado por cuatro capítulos:**

### **Capítulo I: Fundamentación Teórica**

Se abordan los diferentes conceptos asociados al dominio del problema que posibilitan una mejor comprensión y un completo entendimiento de lo planteado en la situación problemática y el marco del problema en sentido general. Se enuncian y argumentan características de algunas aplicaciones similares existentes a nivel mundial que aportan información sobre los sistemas informáticos de personalización de la interfaz gráfica de las aplicaciones *web*, así como la metodología de desarrollo de *software*, el lenguaje de modelado y la herramienta CASE que se utilizarán durante la realización del trabajo para dar cumplimiento al objetivo del mismo.

### **Capítulo II: Caracterización del sistema propuesto**

El capítulo II pretende analizar el contexto donde se desarrollará la solución propuesta. La conceptualización del modelo de dominio permitió obtener un mejor conocimiento sobre el entorno de trabajo donde se desarrolla el módulo, se tuvieron en cuenta cada una de las entidades y conceptos presentes que están relacionados con la herramienta y la relación que existen entre cada uno de estos elementos. Se definen los requisitos funcionales y no funcionales, se construye el diagrama de casos de uso del sistema y se describen los casos de uso del sistema.

### **Capítulo III: Presentación de la solución propuesta**

Se aborda la arquitectura de *software* y los patrones arquitectónicos y de diseño que se utilizarán en el diseño e implementación de la solución propuesta, modelándose la misma a través de los diagramas de clases del diseño y colaboración.

### **Capítulo IV: Implementación y Prueba**

Se exponen los flujos de trabajo de las fases implementación y prueba, donde se desarrolla el diagrama de componentes que describe los elementos físicos del sistema y sus relaciones. Mediante el diagrama de despliegue se muestran los diferentes nodos que componen el sistema y por último y no menos importante se aplican las pruebas a la aplicación para validar las entradas y salidas de la solución y comprobar que cumpla los requerimientos del cliente.

## Capítulo I: Fundamentación Teórica

### 1.1. Introducción

El capítulo I aborda los diferentes conceptos asociados al dominio del problema que posibilitan una mejor comprensión y un completo entendimiento de lo planteado en la situación problemática y el marco del problema en sentido general. Se enuncian y argumentan características de algunas aplicaciones similares existentes a nivel mundial que aportan información sobre los sistemas informáticos de personalización de la interfaz gráfica de las aplicaciones *web*, así como la metodología de desarrollo de *software*, el lenguaje de modelado y la herramienta CASE<sup>1</sup> que se utilizarán durante la realización del trabajo para dar cumplimiento al objetivo del mismo.

### 1.2. La personalización de aplicaciones *web* en el mundo

Han pasado más de 20 años desde que se creó y publicó la primera página *web*. Aunque en aquellos momentos la forma de acceder a ella debía ser con previo conocimiento de la dirección en la que estaba ubicada. Pasados esos años la realidad es otra, aunque muchas de las características que poseían aquellas primeras innovaciones se mantienen. Es evidente el desarrollo alcanzado hoy día, la creación de aplicaciones *web* y más aún la notable evolución tecnológica que alcanza Internet.

Son muchas las instituciones y personas que hacen uso de las tecnologías de la información, más aún de la *web* e Internet. ¿Qué sería el mundo sin la *web* o sin una página *web*? Claramente, las empresas, los bancos, incluso gobiernos dejarían de tener acceso a la información en tan poco tiempo, muchas empresas fracasarían pues la mayoría hacen uso de las Tecnologías de la Información y las Comunicaciones, específicamente de Internet, para promocionar y vender sus productos, las personas no contarían hoy con la posibilidad de buscar y acceder a información del mundo en segundos, sino que tuvieran que acudir a métodos que hoy parecen de varios siglos atrás.

Una novedad actual es la personalización de aplicaciones *web*, alcanzando un desarrollo extraordinario en el mundo. Ejemplos que corroboran lo afirmado anteriormente son las disímiles aplicaciones desarrolladas en Internet brindando servicios online a personas que pueden acceder en

---

<sup>1</sup> Ingeniería de Software Asistida por Computadora.



cualquier momento y desde diferentes lugares. Estas aplicaciones se crean con el objetivo de ayudar a los usuarios a mejorar y satisfacer sus necesidades de una manera más fácil y cómoda. Entre ellas se encuentran: *Artisteer*, *Web Easy*, *Microsoft Expression*, entre otras.

En Cuba, el tema de la personalización de las aplicaciones *web* es nuevo todavía. Las personalizaciones se elaboran según las solicitudes realizadas por los usuarios que interactúan con los sistemas. Los actuales sistemas *web* deben hacer frente al requisito fundamental de adecuarse a las necesidades y preferencias del usuario. Este requisito aparece tanto en sistemas comerciales, con el fin de atraer más clientes potenciales que la competencia, como en sistemas no comerciales, con distintos fines como, por ejemplo, mejorar la accesibilidad del sistema para personas discapacitadas. Son pocos los centros que se dedican a este tipo de actividades.

Estas experiencias se han inclinado más para el sector de la educación, donde más de una universidad ha optado por utilizar dichas personalizaciones como medio de enseñanza para elevar el conocimiento de sus estudiantes y profesores. Algunos de los centros que emplean estas prácticas son la Universidad de La Habana, la Universidad de Pinar del Río, la Universidad de las Ciencias Informáticas (UCI) que cuenta con dos *web* personalizadas: el Entorno Virtual de Aprendizaje (EVA) y el Entorno Virtual de Postgrado que son utilizadas por los estudiantes y profesores para satisfacer sus necesidades de conocimiento. Esta última universidad tiene entre sus líneas de trabajo la personalización, extensión y desarrollo de módulos para el proyecto Moodle. (González, y otros, 2010)

### 1.3. Conceptos Asociados al Dominio del Problema

Para la correcta comprensión de la presente investigación, se hace necesario conocer y dominar los aspectos más significativos relacionados con el objeto de estudio de la misma: los procesos de personalización de la interfaz gráfica de las aplicaciones *web*. Para el total cumplimiento de lo antes planteado ha sido redactado este epígrafe, en el que se exponen los conceptos y elementos que constituyen la base teórica para la realización del presente trabajo de diploma y lógicamente para dar solución al problema científico planteado.

### 1.3.1. VideoWeb

La Plataforma VideoWeb es un producto destinado a la gestión y transmisión de contenido audiovisual a través de la red de datos utilizando tecnología *streaming*, pensada para universidades y otras instituciones que deseen hacer llegar materiales audiovisuales a los usuarios de la red.

### 1.3.2. Aplicación Informática

Una aplicación informática es un *software* que permite al usuario realizar uno o más tipos de trabajo. Los procesadores de texto y las hojas de cálculo son ejemplos de aplicaciones informáticas, mientras que los sistemas operativos o los programas de utilidades que cumplen tareas de mantenimiento no forman parte de estos programas. Las aplicaciones pueden haber sido desarrolladas a medida para satisfacer las necesidades específicas de un usuario o formar parte de un paquete integrado como el caso de Microsoft Office. Las aplicaciones ayudan a los usuarios a lograr sus propósitos y hacen que su trabajo con las mismas sea más placentero y menos penoso. (Definicion.de, 2003)

### 1.3.3. Web

Vocablo inglés que significa “red”, es usado para nombrar a una red informática. Es el componente más usado en Internet. Es usado además para referirse a términos como página *web*, sitio *web*, servidor *web*. (Definicion.de, 2003)

### 1.3.4. Personalización

La personalización es un proceso informático que requiere tiempo y dedicación mediante el cual se pueden lograr cuantiosos avances en la economía de las empresas y el desarrollo intelectual de las personas. Algunos conceptos asociados son:

- ✓ Dar carácter personal a algo.
- ✓ Adaptar las características de un objeto, al gusto o a las necesidades de una persona. (RAE)

Proceso mediante el cual los usuarios podrán modificar las aplicaciones para satisfacer sus gustos y necesidades individuales, es usado por las grandes compañías para hacer publicidad mediante campañas en Internet basadas en *banners*, patrocinios o sofisticados programas de afiliación con el objetivo de conseguir visitas, tráfico, clientes potenciales que promuevan un gran aumento en la economía de las empresas.

#### 1.4. Soluciones o sistemas similares a nivel mundial

En la actualidad, los avances tecnológicos han propiciado el desarrollo de muchas ciencias que ayudan al desarrollo científico de la sociedad. La informática es una de las más favorecidas, se han creado sistemas operativos, programas y herramientas que son usados en la vida cotidiana de los seres humanos. En el mercado internacional pueden encontrarse aplicaciones que poseen algunas funcionalidades similares a la solución propuesta, las mismas son aplicaciones privativas que se venden a un elevado costo y que en su mayoría requieren de un hardware específico para la solución. A continuación se presentan algunas de las características más relevantes de algunos de los sistemas existentes en la actualidad.

##### 1.4.1. Artisteer

*Artisteer* es un diseñador *web* automatizado que permite la creación de sitios *web* completos, *blogs* y además editar las páginas en cuanto al estilo y apariencia de la misma según los gustos de los usuarios. Es una aplicación para *Windows*, permite generar con facilidad un alto nivel en el diseño de plantillas *Web* tanto para Joomla<sup>2</sup>, Wordpress<sup>3</sup>, Drupal<sup>4</sup>, aplicaciones .Net<sup>5</sup> y sitios *web* estáticos sin necesidad de tener conocimientos técnicos sobre XHTML<sup>6</sup> y CSS<sup>7</sup>. El usuario puede modificar la estructura, la tipografía, texturas, imágenes, menús, exportar plantillas. Brinda a los usuarios una

---

<sup>2</sup> Sistema de gestión de contenidos que permite integrar, añadir o editar el contenido de un sitio web de manera sencilla.

<sup>3</sup> Sistema de gestión de contenido enfocado a la creación de sitios web periódicamente actualizados.

<sup>4</sup> Sistema de gestión de contenido modular que permite publicar artículos, imágenes, u otros archivos.

<sup>5</sup> Framework para aplicaciones web desarrollado y comercializado por Microsoft.

<sup>6</sup> Adaptación de HTML al lenguaje XML.

<sup>7</sup> Lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

opción denominada “Sugerir” con la cual el programa crea un diseño al azar y puede ser usado en beneficio de los mismos. Es prácticamente un editor WYSIWYG<sup>8</sup> para plantillas. (Domínguez, 2005)

Todo lo que en él se desarrolla es completamente visual, no es necesario introducir código, tiene una interfaz simple e intuitiva, es una aplicación fácil de usar y además presenta soporte en diferentes idiomas, presenta buena compatibilidad con los sistemas operativos *Windows – Windows XP / Windows 2000 / Windows Vista / Windows 7*. Para poder utilizarlo es necesario como mínimo: Procesador: Pentium III 500 MHz, Memoria: 128 MB de RAM, espacio libre en disco: 80 MB, aunque es más recomendable en un ordenador con: Procesador: 2.4 GHz Pentium IV, Memoria: 512 MB RAM, espacio libre en disco: 580 MB. (Domínguez, 2005)

#### 1.4.2. Microsoft Expression

*Microsoft Expression* es una combinación entre un visualizador y un catalogador multimedia, donde se armoniza el carácter funcional con el concepto de profesionalidad para responder a todas las necesidades de los usuarios. Es una familia de herramientas para diseñadores profesionales para la creación de aplicaciones *web* y aplicaciones clientes de *Windows*, así como contenido multimedia sofisticado. Es un programa que abre un nuevo mundo de posibilidades creativas y las herramientas de diseño ofrecen a los usuarios la libertad necesaria para que la visión de sus aplicaciones se haga realidad. (Djthreem, 2011)

#### 1.4.3. Web Easy

*Web Easy Professional* es un editor de páginas *web* bastante completo enfocado a usuarios inexpertos sin ningún conocimiento de HTML que quieran hacer una página *web* desde la vista diseño. Es muy particular y con una interfaz poco habitual entre los editores comúnmente utilizados por desarrolladores avanzados, presenta más de 500 plantillas de sitios *web* para personalizar, permite la optimización del sitio *web* y ofrece a los usuarios una gama de productos para elegir la solución que mejor se adapte a sus necesidades, genera álbumes de fotos virtuales e incluye más de 85.000 gráficos listos para insertar en los sitios *web* que se crearán, entre ellos botones, fondos, fotografías, marcos y atractivas animaciones. (Alvarez, 2003)

---

<sup>8</sup> Herramienta que permite crear elementos de formulario donde el usuario puede escribir texto con estilos.

Luego de haber realizado un detallado estudio sobre algunos de los sistemas similares existentes y más usados en el mundo, analizar sus ventajas, desventajas y las características más importantes de cada uno es necesario resaltar que son soluciones muy utilizadas actualmente y muy potentes. Valorando críticamente estos sistemas se observa que en su mayoría son soluciones propietarias.

Son sistemas que permiten a los usuarios crear las plantillas totalmente diferentes y con un alto nivel de desarrollo sin tener previos conocimientos de programación, permite entre otras cosas modificar la fuente, el banners, los botones, los colores de fondo, cambiar la apariencia. Hoy día el país está preparando poco a poco a los usuarios para que el cambio de *software* propietario a *software* libre no sea tan brusco.

La UCI es uno de los centros desarrolladores de *software* que está mayormente enfrascado en el proceso de migración hacia el *Software* Libre. El trabajo con tecnología libre brinda a los desarrolladores un sinnúmero de ventajas, estos *software* son más económicos, tienen libertad de uso y pueden ser redistribuidos, mientras que los *software* propietarios no tienen aplicaciones para todas las plataformas, el código no puede ser copiado ni modificado, el soporte de la aplicación es exclusivo del propietario, el coste de las aplicaciones es mayor además de que el usuario que adquiera el *software* propietario depende al 100% de la empresa propietaria. En la Universidad se han impartido cursos y conferencias de preparación al personal para el trabajo con este sistema operativo. Se recomienda, por todas las ventajas que brinda el sistema operativo Linux, la utilización de herramientas libres o multiplataforma para el desarrollo informático de las aplicaciones futuras.

### 1.5. Metodología de Desarrollo del *Software*

Con el objetivo de mantener el control, la estabilidad y la organización en un proyecto se deben definir desde un principio los pasos a seguir para el desarrollo del mismo, donde se especifiquen las actividades, los controles y la documentación realmente necesaria para el equipo de desarrollo del proyecto y el producto a realizar. Es por ello que el proceso de selección de la metodología a aplicar para la realización del proyecto fue riguroso.

Las Metodologías de Desarrollo de *Software* son un conjunto de procedimientos, técnicas, herramientas y soporte documental en el desarrollo de un producto *software*, especifican la información necesaria para comenzar una actividad y la información que se debe producir como resultado de la

misma. Las metodologías definen que hacer, cómo y cuándo durante todo el desarrollo y mantenimiento de un proyecto. Se clasifican en estructuradas y no estructuradas. (Carrillo Pérez, y otros, 2008)

El uso de las metodologías permite mayor rapidez y que los resultados que se obtengan sean del agrado de los clientes, ayuda a los ingenieros a comprender mejor el problema, facilita la evaluación de los resultados y el cumplimiento de los objetivos, permite la reutilización de las partes del producto y garantiza a los usuarios un determinado nivel de calidad en el producto final. A la hora de elegir una metodología de desarrollo para la realización de un producto es necesario tener en cuenta las características que presenta el equipamiento con que se va a desarrollar y las del proyecto que se va a ejecutar.

### 1.5.1. Proceso Unificado de Desarrollo (RUP)

Para el desarrollo de la solución propuesta se selecciona RUP<sup>9</sup> como metodología de desarrollo por las características y ventajas que trae consigo trabajar con esta metodología. La misma se adapta a las condiciones del proyecto mediante una previa disposición para su uso, realizando una configuración adecuada que hace que el desarrollo del *software* sea rápido. Permite controlar los posibles cambios que puedan ocurrir durante todo el proceso de desarrollo, define claramente los objetivos de cada fase, los entregables y sus actividades permitiendo un entorno en el que se agilicen las prácticas y técnicas necesarias para llevar a cabo la integración continua manteniendo diseños simples.

La documentación que se genera durante el proceso de desarrollo facilita que el cliente y el grupo de desarrollo del *software* obtengan un mejor entendimiento de la solución, el equipo de trabajo tiene experiencia en el uso de esta metodología, lo que hace que se desarrolle más fácil y rápidamente el producto. Propone varias iteraciones en el proceso de desarrollo de *software* posibilitando la corrección de errores y mejoras del *software* a medida que avanza el proyecto, lo que permite a los desarrolladores poder realizar un producto más completo y con gran calidad. (Gómez Gallego, 2007)

---

<sup>9</sup> Proceso Unificado de Desarrollo

## 1.6. Lenguaje Unificado de Modelado (UML)

Se utiliza para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas *software* como para la arquitectura *hardware* donde se ejecuten. Es un lenguaje que se centra en la representación gráfica de un sistema, es utilizado para especificar, construir, visualizar y documentar los artefactos de un sistema de *software* orientado a objetos. No es un método de desarrollo, lo que significa que no sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema, sino que simplemente ayuda a visualizar el diseño y a hacerlo más accesible para otros. (Cardona, 2011)

Por las características expuestas anteriormente se decide utilizar el lenguaje UML en su versión 2.0 durante el ciclo de vida del proyecto pues da la posibilidad a los analistas de definir las características principales del sistema, describe las decisiones y todos los pasos necesarios representados gráficamente por modelos precisos y completos que proporcionan una mejor comprensión para los desarrolladores. Es un lenguaje que puede ser implementado en cualquier lenguaje orientado a objeto y permite tanto a los clientes como a los desarrolladores saber a ciencia cierta el alcance y la factibilidad que puede tener el producto final.

## 1.7. Herramientas CASE

Las herramientas CASE son la aplicación de tecnología informática a las actividades, técnicas y metodologías de un sistema. Fueron creadas con el objetivo de acelerar, automatizar y apoyar el trabajo que se realice durante el ciclo de vida de un proyecto. Son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de *software* y desarrolladores durante el ciclo de vida de un proyecto, están encaminadas a aumentar la calidad y eficiencia del *software* y de la documentación, propiciando mejoras en la planificación, productividad en el desarrollo y mantenimiento del proyecto, facilitan la automatización del ciclo de vida del desarrollo de sistemas de información completamente o en alguna de sus fases. Estas herramientas permiten a los analistas tener más tiempo para el análisis y diseño de la aplicación además de minimizar el tiempo para codificar y probar la solución. (Latina Osmosis, 2000)

Las metodologías y técnicas soportadas, los lenguajes y sistemas gestores de base de datos a los que va orientada y las posibilidades de integración con otras plataformas son algunos criterios que se deben tener en cuenta en el momento de seleccionar una herramienta CASE para la modelación de

cualquier proyecto o *software*. Juegan un papel importante durante la modelación del proyecto y ayudan en gran medida a los desarrolladores y analistas del mismo pues aumentan la productividad y disminuyen el tiempo necesario para documentar, analizar y desarrollar los sistemas, son esenciales para el análisis de los proyectos, mejoran la forma en que ocurre el desarrollo y tiene influencia sobre la calidad de resultado final. (Giraldo Zapata, 2005)

### 1.7.1. Visual Paradigm

Es una herramienta CASE para realizar el modelado UML siguiendo el estándar UML 2.1. Presenta características gráficas muy cómodas que facilitan la realización de los diagramas de modelado. Permite exportar imágenes con extensión jpg<sup>10</sup> y png<sup>11</sup>. (Valdez Altamirano, 2009). El trabajo con esta herramienta proporciona ciertos beneficios para el equipo de desarrollo, ejemplo de esto es que soporta aplicaciones *web* y varios idiomas, es fácil de instalar y actualizar, tiene un diseño centrado en casos de uso y enfocado al negocio que genera un *software* de mayor calidad, disponibilidad de integrarse en los principales IDEs<sup>12</sup> y en múltiples plataformas, presenta compatibilidad entre ediciones y como resultado brinda un producto de alta calidad. (Giraldo Zapata, 2005)

Es la herramienta CASE que utiliza UML como lenguaje de modelado, está diseñada para manejar grandes y complicadas estructuras de un proyecto de forma eficiente, soporta el ciclo de vida completo del *software*: análisis y diseño, construcción, pruebas y despliegue, permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación, presenta licencia gratuita y comercial. (Valdez Altamirano, 2009)

Debido a la necesidad de migrar a sistemas libres o multiplataforma, se decide utilizar Visual Paradigm para maximizar el trabajo tanto individual como colectivo de los desarrolladores y además porque es completamente compatible con la metodología que se decidió utilizar, brinda muchas facilidades en la generación de la documentación del módulo que se está desarrollando, propiciando un buen desarrollo del producto y una mayor calidad en el *software* final.

---

<sup>10</sup> Formato de compresión de imágenes, tanto en color como en escala de grises, con alta calidad (a todo color).

<sup>11</sup> Formato gráfico basado en un algoritmo de compresión sin pérdida para bitmaps no sujeto a patentes.

<sup>12</sup> Entorno de Desarrollo Integrado.



## 1.8. Lenguajes de Programación

Un lenguaje de programación no es más que un conjunto de símbolos, reglas sintácticas y semánticas, mediante las cuales el programador puede interactuar con el *hardware* para obtener un resultado, las instrucciones que forman dicho resultado son conocidas como código fuente, pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Existen un gran número de lenguajes, pero ninguno se puede considerar el más capacitado o el más adecuado para creación de aplicaciones. (Marin Arias, 2008). En el presente epígrafe se abordan una serie de lenguajes que se utilizarán para dar solución a los objetivos planteados y los indicadores que posibilitaron su selección.

### 1.8.1. HTML

Lenguaje Marcado de Hipertexto o *HyperText Markup Language (HTML)* es un estándar mundialmente utilizado para la elaboración de páginas *web*. Sus normas están definidas por la *World Wide Web Consortium (W3C)* favoreciendo que toda página HTML sea vista de similar manera desde cualquier navegador aunque el sistema operativo utilizado sea diferente.

Este lenguaje puede ser creado y editado usando editores de texto básico (Bloc de notas) aparte de otras aplicaciones que tienen como objetivo el diseño y la construcción de sitios *web (Dreamweaver)*. Es un lenguaje sencillo y presenta el texto de una forma estructurada y agradable en pequeños archivos, sin embargo, tiene como inconveniente que es un lenguaje estático y hace que el diseño sea más lento, por lo que se recomienda que se utilice acompañado de otros lenguajes como JavaScript que ofrecen funcionalidades adicionales. (Politécnica de Catalunya, 2002)

### 1.8.2. PHP

Es un lenguaje de programación interpretado por la mayoría de los servidores *web* incluyendo *Apache* que no necesita ser compilado para ejecutarse. Es especialmente utilizado para la creación de sitios dinámicos incluyendo el código fácilmente dentro del HTML. Está publicado bajo la *PHP License*, que es considerada por la *Free Software Foundation* como *software* libre. (PHP, 2001)

Se ha elegido este lenguaje pues su aprendizaje es fácil y rápido, es multiplataforma, tiene una gran capacidad de conexión con la mayoría de los gestores de bases de datos, incluye gran cantidad de

funciones y no requiere definición de variables. Entre sus habilidades se encuentran la creación de imágenes, archivos PDF e incluso películas Flash.

### 1.8.3. JavaScript

*JavaScript* es un lenguaje de programación "ligero" y orientado a objetos, tiene un magnífico control sobre el navegador ya que es usado para incluir código en el cliente que será interpretado por el navegador. Es el lenguaje de programación *web* del lado del cliente más extendido. Dentro de sus ventajas figura que es un *scripting* seguro y fiable, los *script* tienen capacidad limitada por razones de seguridad y el código se ejecuta en el cliente. Algunas desventajas son que el código debe descargarse completamente y presenta un código visible para cualquier usuario.

## 1.9. Entorno de Desarrollo Integrado (IDE)

Un Entorno Integrado de Desarrollo (*Integrated Development Environment*) es una aplicación compuesta de herramientas útiles para un programador que puede ser usado tanto para un solo lenguaje de programación como para varios. Están compuestos por un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Dentro de los IDEs que soportan múltiples lenguajes se encuentran *NetBeans* y *Eclipse* los cuales son proporcionados por *plugins* que se instalan en el mismo. Su construcción está basada en lenguajes multiplataforma como *Java* o *MonoDevelop* lo cual ofrece como característica que se ejecuten en múltiples plataformas incluyendo *Windows*, *GNU/Linux* y *Mac OS*.

Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Java, C#, Basic, Object Pascal, PHP. Algunos ejemplos de entornos integrados de desarrollo (IDE) son: *Eclipse*, *NetBeans*, *MS Visual Studio .NET* de Microsoft y el *Zend Studio*. (S.A. Dart, 1987)

### 1.9.1. NetBeans 6.9

El IDE *NetBeans* es un reconocido entorno de desarrollo integrado disponible para *Windows*, *Mac*, *Linux* y *Solaris*. El proyecto *NetBeans* está formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones *web*, empresariales, de

escritorio y móviles utilizando la plataforma Java, PHP<sup>13</sup>, JavaScript y Ajax, entre otras. El proyecto de *NetBeans* está apoyado por una comunidad de desarrolladores dinámica y ofrece documentación y recursos de formación exhaustivos. (NetBeans Team, 2011)

Se determinó utilizar el IDE *NetBeans* 6.9 pues respecto a otros IDE tiene un mejor aprovechamiento del tiempo de desarrollo, alta organización de la aplicación basada en estándares y patrones estructurales y de diseño. Posee una arquitectura consistente y robusta y un mejor rendimiento en cuanto a tiempo de ejecución y optimización de recursos, lo que permitirá a los desarrolladores la creación de un producto completo.

### 1.10. Tecnologías para el Desarrollo de Aplicaciones

Existen disímiles formas para desarrollar una aplicación *web*. El empleo de herramientas como *Frameworks* o *CMS*<sup>14</sup> posibilita un gran punto de partida pues tienen por lo general una estructura sólida que agiliza el proceso de construcción y permiten desarrollar aplicaciones complejas en un tiempo relativamente más rápido que cuando se realiza desde cero.

#### 1.10.1. CMS Drupal 6.14

*Drupal* es un sistema de gestión de contenido modular y muy configurable. Es un programa de código abierto, con licencia *GNU/GPL*, escrito en *PHP*, desarrollado y mantenido por una activa comunidad de usuarios. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la *web*, y un énfasis especial en la usabilidad y consistencia de todo el sistema.

El diseño de *Drupal* es especialmente idóneo para construir y gestionar comunidades en Internet. No obstante, su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar diferentes tipos de sitios *web*. Permite publicar artículos, imágenes u otros archivos y servicios añadidos como foros, encuestas, votaciones, *blogs* y administración de usuarios y permisos. (Paredes García, 2010)

---

<sup>13</sup> Lenguaje de programación interpretado (Lenguaje de alto rendimiento), diseñado originalmente para la creación de páginas web dinámicas.

<sup>14</sup> Sistemas de Gestión de Contenidos.

Drupal está desarrollado usando el lenguaje de programación *PHP*, lenguaje con el que el equipo de desarrollo está familiarizado, lo que posibilita y agiliza el desarrollo y adaptación de componentes para el funcionamiento de la plataforma VideoWeb. Las funcionalidades que brinda este *CMS* son muchas. Drupal además de soportar *MySQL*<sup>15</sup> como Sistema Gestor de Base de Datos también permite el uso de *PostgreSQL*, cuenta con gestión de privilegios granulares, lo que posibilita una mayor especificación del acceso a los recursos. (Torres Rodríguez, 2009). En la Universidad de las Ciencias Informáticas existen diferentes proyectos que desarrollan portales usando el *CMS Drupal*, lo que posibilita el apoyo e intercambio de experiencias facilitando la preparación del equipo de desarrollo.

### 1.11. Servidor Web Apache 2.2

Desde 1996 es el servidor *HTTP*<sup>16</sup> más usado, es una de las aplicaciones estrella del mundo del *software* libre ya que es el servidor *web* de mayor implantación, está disponible para *GNU/Linux*, *Solaris*, *Microsoft Windows* y otras plataformas minoritarias. Apache es flexible, rápido y eficiente y continuamente se está actualizando, es multiplataforma y además puede ser adaptado a diferentes entornos y necesidades. La arquitectura utilizada es cliente/servidor, es decir, el equipo cliente hace una solicitud o petición al equipo servidor y éste la atiende. En el equipo cliente se ejecuta una aplicación llamada “navegador o cliente *web*” que sirve de interfaz con el usuario, atiende sus peticiones, muestra los resultados de las consultas y proporciona un conjunto de herramientas que facilitan su comunicación con el servidor. El protocolo utilizado para la transferencia de hipertexto es *HTTP* que está basado en el envío de mensajes y establece el conjunto de normas mediante las cuales se envían las peticiones de acceso a una *web* y la respuesta de esa *web*. (Ubuntu, 2011)

### 1.12. Conclusiones

El presente capítulo aborda un estudio detallado de las definiciones de los conceptos básicos relacionados con el dominio del problema, contribuyendo al enriquecimiento del desarrollo de la investigación logrando un mayor entendimiento de esta. Se caracterizan varias de las soluciones similares existentes en el mundo que se utilizan para personalizar las aplicaciones *web*, se llegó a la

---

<sup>15</sup> Sistema de gestión de bases de datos relacional más usado en el mundo del *software* libre, debido a su gran rapidez y facilidad de uso.

<sup>16</sup> Protocolo de transferencia de hipertexto. Es el protocolo usado en cada transacción de la *World Wide Web*.

conclusión de que no pueden ser utilizadas porque son propietarias y las que se requieren deben ser libres y multiplataforma.

Se caracterizó además los lenguajes de programación que se utilizan para desarrollar la aplicación. De las metodologías de desarrollo existentes, se decide utilizar *RUP* por las facilidades que brinda y la excelente organización en la documentación del trabajo. El lenguaje de modelado elegido es *UML* y para la realización de los diagramas necesarios se eligió la herramienta *CASE Visual Paradigm*. Se caracterizó las tecnologías, herramientas y tendencias asociadas a la posible propuesta para solucionar el problema científico que concierne la presente investigación.

## Capítulo II: Caracterización del Sistema Propuesto

### 2.1. Introducción

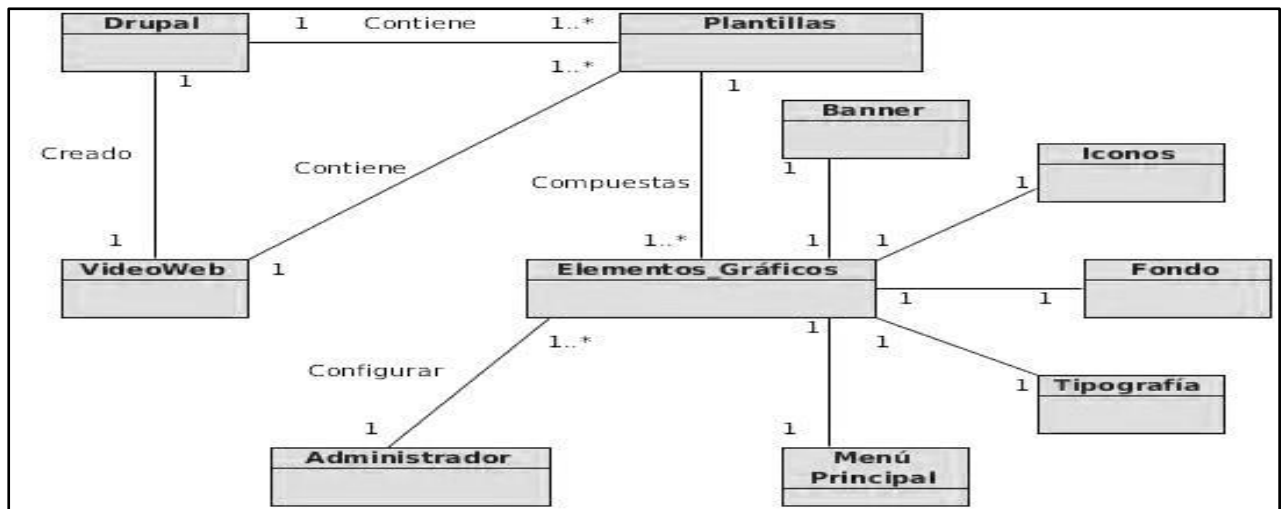
El capítulo II pretende analizar el contexto donde se desarrollará la solución propuesta. La conceptualización del modelo de dominio permitió obtener un mejor conocimiento sobre el entorno de trabajo donde se desarrolla el módulo, se tuvieron en cuenta cada una de las entidades y conceptos presentes que están relacionados con la herramienta y la relación que existentes entre cada uno de estos elementos. Se definen los requisitos funcionales y no funcionales, se construye el diagrama de casos de uso del sistema y se describen los casos de uso del sistema.

### 2.2. Modelo de Dominio

Teniendo en cuenta que el producto que se desea desarrollar está concebido como una herramienta de personalización para las aplicaciones *web* en la Plataforma VideoWeb, y el comportamiento de la misma está sujeto a una diversidad de cambios, se hace difícil encontrar procesos de negocios bien estructurados que permitan realizar un completo modelado de dicho negocio, por lo que se decide realizar un modelo de dominio.

Un modelo de dominio es un diagrama con los objetos reales existentes relacionados con el proyecto que se va a realizar y las relaciones que hay entre ellos. El objetivo de este modelo es ayudar a comprender los conceptos que utilizan los usuarios, con los que se trabaja y los conceptos con los que deberá trabajar la aplicación. Puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema. Es utilizado por el analista como un medio para comprender el sector de negocios al cual el sistema va a servir.

El modelo de dominio puede ser tomado como el punto de partida para el diseño del sistema. Cuando se realiza la programación orientada a objetos, el funcionamiento interno del *software* va a imitar en alguna medida a la realidad, por lo que el mapa de conceptos del modelo de dominio constituye una primera versión del sistema. (Larman, 1999)



**Figura 1: Diagrama de Clases del Dominio**

### 2.3. Breve descripción del Diagrama de Clases del Dominio

Las siguientes definiciones ayudarán los usuarios a tener una mejor comprensión de los conceptos relacionados con el dominio del problema:

- ✓ **VideoWeb:** La Plataforma VideoWeb es un producto destinado a la gestión y transmisión de contenido audiovisual a través de la red de datos utilizando tecnología *streaming*.
- ✓ **Drupal:** Herramienta potente y flexible que permite agilizar el desarrollo y adaptación de componentes para el funcionamiento de la Plataforma VideoWeb.
- ✓ **Plantillas:** Estructura que permitirá visualizar los cambios que lleve a cabo el usuario administrador en la página principal de la plataforma.
- ✓ **Elementos Gráficos:** Son aquellos elementos estructurales que podrán sufrir modificaciones, es decir, pueden ser personalizados por el usuario administrador.
- ✓ **Administrador:** Usuario responsable de los cambios o transformaciones realizadas en la plataforma.
- ✓ **Menú Principal:** Permite al usuario desplazarse por todos los elementos de la *web* en forma de árbol, es decir, brindar información del producto, búsquedas y páginas de contacto.
- ✓ **Fondo:** Color o imagen que en la página *web* se podrá observar en el último plano.
- ✓ **Banner:** Sección superior de la página donde se muestra por lo general el nombre de la misma.
- ✓ **Tipografía:** Información de la página disponible en la plataforma que podrá ser modificada por el usuario administrador mediante las disímiles opciones que se brindan.

- ✓ **Íconos:** Son aquellos elementos que estarán disponibles para que el usuario administrador pueda cambiar la iconografía de la plantilla.

## **2.4. Especificación de los Requisitos del Sistema**

El *software* a desarrollar tiene que cumplir una serie de requisitos, que describirán el comportamiento del sistema, vinculado a los servicios que el usuario desea que brinde y las restricciones bajo las cuales el mismo debe operar. Atendiendo a las necesidades de personalización que tienen actualmente la plataforma y partiendo de que el presente trabajo es la continuación del Trabajo de Diploma “Análisis de un sistema de personalización para las aplicaciones *web* en la Plataforma de Transmisión Abierta para Radio y Televisión”, se hace necesario revisar a fondo la ingeniería del mismo para definir cuáles son los requisitos del sistema que se mantienen en la fase de implementación.

Luego de haber realizado un extenso análisis de la ingeniería realizada en el trabajo de diploma mencionado en el párrafo anterior se llegó a la conclusión de que los artefactos generados debían ser cambiados y ajustados hasta satisfacer las necesidades de los usuarios.

### **2.4.1. Requisitos Funcionales**

Son declaraciones de los servicios que debe proporcionar el sistema, especifican la manera en que éste debe reaccionar a determinadas entradas, cómo debe comportarse el sistema en situaciones particulares, y además pueden declarar explícitamente lo que el sistema no debe hacer. (Linares Armas, 2008)

#### **1. RF1 Modificar Fondo Página**

El sistema debe permitir al administrador poder cambiar el color de fondo de la página.

#### **2. RF2 Modificar Banner**

El sistema debe permitir al administrador poder realizar diferentes cambios sobre el encabezado de la página.

##### ✓ **RF2.1 Cambiar color de fondo del banner**

El sistema debe permitir al administrador cambiar el color de fondo que tenga el encabezado seleccionando otro color.



✓ **RF2.2 Imagen para fondo del banner**

El sistema debe permitir al administrador insertar desde el archivo una imagen en el encabezado.

**3. RF3 Modificar Menú Principal**

El sistema debe permitir al administrador configurar el menú principal en cuanto a tipo de fuente, tamaño y color que prefiera.

**4. RF4 Modificar Tipografía**

El sistema debe permitir al administrador modificar la información de la página en cuanto a fuente, tamaño de la fuente y color.

**5. RF5 Cambiar Iconografía**

El sistema debe permitir al administrador cambiar el logo del banner por otro.

**6. RF6 Guardar Cambios**

El sistema debe permitir al administrador salvar la nueva configuración realizada.

### 2.3.2. Requisitos No Funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema, son propiedades o cualidades que el producto debe tener, las características que hacen al producto atractivo, usable, rápido o confiable. No son parte de la razón fundamental del producto pero si son necesarios para hacer funcionar el producto de la manera deseada. Surgen de las necesidades del usuario, debido a las restricciones en el presupuesto, a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas *software* o *hardware*, debido a factores externos como regulaciones de seguridad o legislaciones sobre privacidad. Se conocen como un conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar el *Software*. (Pressman, 2005)

**1. RNF1 Apariencia o interfaz gráfica**

Debe mostrar una interfaz gráfica agradable al usuario, así como un diseño sencillo que proporcione de forma fácil cualquier interacción con el sistema.

**2. RNF2 Usabilidad**

El sistema debe ser de fácil uso para todo el personal, tanto para las personas que tengan conocimientos mínimos de computación como para las inexpertas.

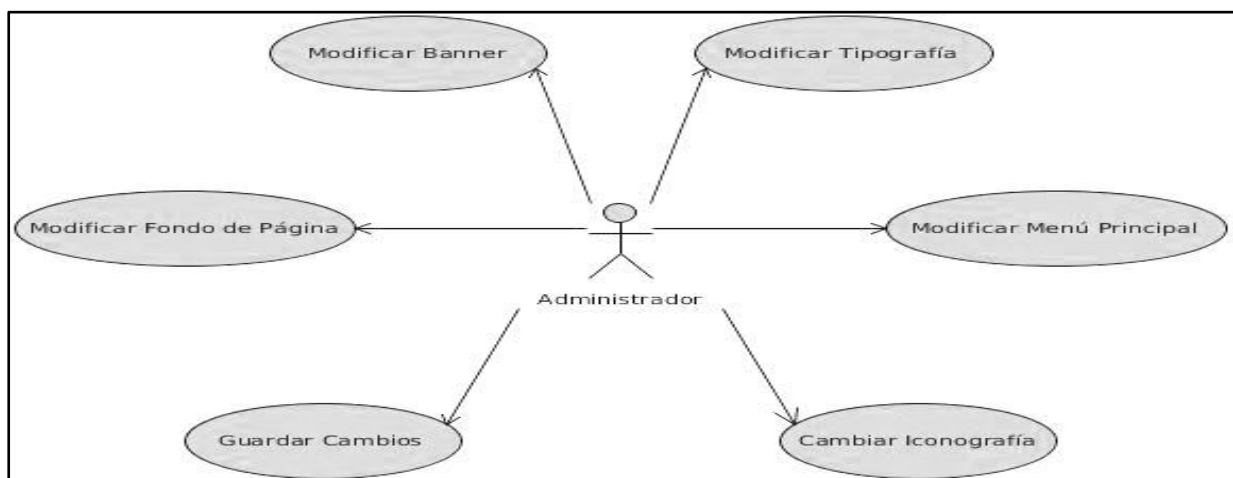
### 3. RNF3 Requisitos de Hardware

- ✓ Tarjeta de red.
- ✓ Se requiere al menos 100 MB de disco duro.
- ✓ Al menos 256 MB de memoria RAM.
- ✓ El procesador del sistema debe tener una velocidad de al menos 1.6GHz

### 2.5. Diagrama de Casos de Uso del Sistema

El modelo de Casos de Uso del Sistema describe las funcionalidades propuestas del nuevo sistema. Estas funcionalidades se relacionan con actores hasta conformar diagramas. Un diagrama puede estar compuesto por uno o más actores vinculados con uno o más casos de uso. Los casos de usos ayudan a comunicar de manera eficiente las funciones del sistema al cliente, para lograr que esta comunicación sea lo más eficiente posible los casos de uso deben describirse utilizando el lenguaje del cliente.

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario, por lo tanto, determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar. Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente. Un caso de uso debe especificar un comportamiento deseado, pero no imponer cómo se llevará a cabo ese comportamiento, es decir, debe decir QUÉ pero no CÓMO. (Cáceres Tello, 2009)



**Figura 2: Diagrama de Casos de Uso del Sistema**



### 2.5.1. Descripción de los Casos de Uso del Sistema

La descripción de un Caso de Uso generalmente incluye notas que lo describen y comentarios generales tales como:

- ✓ **Requisitos:** Acciones que el CU debe permitir hacer al usuario.
- ✓ **Restricciones:** Incluye precondiciones que tienen que ser verdaderas antes que se ejecute el CU y poscondiciones que deben ser verdaderas una vez que se ejecute el mismo. Reglas sobre ¿qué acción se puede o no hacer?
- ✓ **Escenarios:** Descripciones secuenciales que se llevan a cabo durante la ejecución del CU, los cuales pueden incluir múltiples escenarios.

#### Modificar Fondo Página

<b>Caso de Uso</b>	Modificar Fondo Página
<b>Actor</b>	Administrador
<b>Propósito</b>	Este caso de uso tiene como objetivo que el administrador pueda cambiar el fondo de la página por otro deseado.
<b>Resumen</b>	El caso de uso se inicia cuando el administrador desea cambiar el fondo que tiene la página y selecciona la opción Fondo, finalizando el mismo cuando el fondo de la página es cambiado.
<b>Precondiciones</b>	El administrador debe estar autenticado.
<b>Referencias</b>	RF1
<b>Prioridad</b>	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona la opción "Fondo".	1.1. El sistema muestra una interfaz donde se encuentran las opciones siguientes: A. Color de Fondo. B. Imagen de Fondo.
2. El administrador selecciona la opción A).	2.1. Ver sección (A) "Color de Fondo".
3. El administrador selecciona la opción B).	3.1. Ver sección (B) "Imagen de Fondo".

	4. El sistema procesa la información a partir de la acción realizada por el administrador y actualiza la visualización en el fondo de la página.
<b>Sección (A) “Color de Fondo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador selecciona la opción “Color”.	1.1. El sistema da la posibilidad al administrador de seleccionar el color de relleno que desea aplicar al fondo de la página.
2. El administrador selecciona el color que desea.	2.1. El sistema visualiza en el Fondo de Página el color seleccionado por el administrador.
<b>Flujo Alternativo</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El administrador no selecciona ningún color.	2.1. El sistema no realiza ninguna transformación.
<b>Prototipo de Interfaz</b>	
	
<b>Sección (B) “Imagen de Fondo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador selecciona la opción “Imagen”.	1.1. El sistema da la posibilidad al administrador de buscar una imagen cualquiera para aplicar al fondo de la página.
2. El administrador selecciona la imagen que desea.	1.1. El sistema visualiza en el Fondo de Página la imagen seleccionada por el administrador.
<b>Flujo Alternativo</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El administrador no selecciona ninguna imagen.	2.1. El sistema no realiza ninguna transformación.
<b>Prototipo de Interfaz</b>	
	

<b>Poscondiciones</b>	Se modifica el Fondo de Página y se visualiza el color o la imagen seleccionado por el administrador.
-----------------------	---

**Tabla 1: Descripción del CU Cambiar Fondo de Página**

**Modificar Banner**

<b>Caso de Uso</b>	Modificar Banner
<b>Actor</b>	Administrador
<b>Propósito</b>	Este caso de uso tiene como objetivo brindar al administrador la posibilidad de modificar el encabezado o <i>banner</i> de la página.
<b>Resumen</b>	El caso de uso se inicia cuando el administrador desea modificar el <i>banner</i> de la página y selecciona la opción <i>Banner</i> , finalizando el mismo cuando el administrador obtiene las modificaciones deseadas.
<b>Precondiciones</b>	El administrador debe estar autenticado.
<b>Referencias</b>	RF2
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador selecciona la opción " <i>Banner</i> ".	1.1. El sistema muestra una interfaz donde brinda la opción para modificar el <i>banner</i> de la página: A. Imagen. B. Color de Fondo.
2. El administrador selecciona la opción A).	2.1. Ver sección (A) "Imagen".
3. El administrador selecciona la opción B).	3.1. Ver sección (B) "Color de Fondo".
	4. El sistema procesa la información a partir de la acción realizada por el administrador y actualiza la visualización en el <i>banner</i> de la página.
<b>Prototipo de Interfaz</b>	

Banner

Color de Fondo:

Imagen de fondo:  
   
 Seleccione la imagen de fondo que desea para el banner.

Logo de la página:  
   
 Seleccione el logo que desea para el banner.

**Sección (A) "Imagen"**

Acción del Actor	Respuesta del Sistema
	1.1. El sistema brinda la posibilidad de insertar una imagen desde un archivo.
2. El administrador selecciona la imagen que desea tener en la página.	2.1. El sistema visualiza en la página la imagen seleccionada por el administrador.

**Flujo Alterno**

Acción del Actor	Respuesta del Sistema
2. El administrador no selecciona ninguna imagen para insertar en el banner de la página.	2.1. El sistema no realiza ninguna transformación.


**Prototipo de Interfaz**

Imagen de fondo:  
   
 Seleccione la imagen de fondo que desea para el banner.

**Sección (B) "Color de Fondo"**

Acción del Actor	Respuesta del Sistema
	1.1. El sistema brinda la posibilidad de seleccionar un color para cambiar el color de fondo de la página.
2. El administrador selecciona el color que desea tenga el banner de la página.	2.1. El sistema visualiza en la página el nuevo color del banner seleccionado por el administrador.


**Flujo Alterno**

Acción del Actor	Respuesta del Sistema
2. El administrador no selecciona ningún color para aplicar al banner de la página.	2.1. El sistema no realiza ninguna transformación.
<b>Prototipo de Interfaz</b>	
<p>Color de Fondo:</p> 	
<b>Poscondiciones</b>	Se modifica el banner de la página teniendo en cuenta la selección del administrador.

**Tabla 2: Descripción del CU Modificar Banner**

### Modificar Menú Principal

<b>Caso de Uso</b>	Modificar Menú Principal
<b>Actor</b>	Administrador
<b>Propósito</b>	Este caso de uso tiene como objetivo que el administrador pueda configurar el menú principal de la página.
<b>Resumen</b>	El caso de uso se inicia cuando el administrador desea configurar el menú principal de la página y selecciona la opción Menú Principal, finalizando el mismo cuando el menú queda configurado.
<b>Precondiciones</b>	El administrador debe estar autenticado.
<b>Referencias</b>	RF3
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona la opción "Menú Principal".	<p>1.1. El sistema muestra una interfaz donde brinda la opción de configurar el menú de la página mediante:</p> <ul style="list-style-type: none"> <li>• Formato del texto: El sistema muestra unas listas desplegables para que el administrador seleccione el color, la familia de fuente y el tamaño que desea aplicar al menú principal de la página.</li> </ul>

2. El administrador selecciona las transformaciones que desea aplicar al Menú Principal.	2.1. El sistema procesa la información a partir de la acción realizada por el administrador y actualiza la visualización en el menú principal.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
2. El administrador no selecciona ninguna de las opciones de transformación para aplicar al Menú Principal.	2.1. El sistema no realiza ninguna transformación
Prototipo de Interfaz	
	
Poscondiciones	Se configura el menú de navegación de la página y se visualiza todos los cambios realizados teniendo en cuenta las opciones seleccionadas.

**Tabla 3: Descripción del CU Configurar Menú Principal**

## 2.6. Conclusiones

En este capítulo se abordó la conceptualización del entorno de trabajo mediante el Modelo de Dominio, en el cual se analizaron cada una de las entidades y conceptos presentes en el contexto de desarrollo de la herramienta, además de las relaciones existentes entre cada uno de estos elementos. Se definió los requisitos del sistema, tanto los funcionales como los no funcionales que deberá cumplir la solución propuesta. También se presentó el diagrama de Casos de Uso del Sistema con una descripción textual de los Casos de Uso y Actores que en el intervienen.



## Capítulo III: Análisis y Diseño del Sistema

### 3.1. Introducción

El capítulo III aborda la arquitectura de *software* y los patrones arquitectónicos a utilizar en el diseño e implementación de la solución propuesta, modelándose la misma a través de los diagramas de clases del análisis, diseño e interacción.

### 3.2. Patrón Arquitectónico

En un sistema informático la arquitectura es el conjunto de decisiones significativas sobre la organización de un sistema *software*. Para describir y diferenciar una arquitectura se hace necesario emplear estilos arquitectónicos que ayuden a definir una estructura para todos los componentes del sistema.

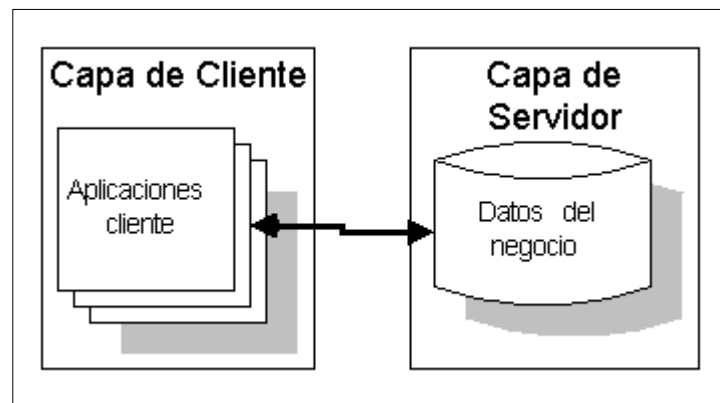
Los patrones arquitectónicos definen la estructura general del *software*, indican las relaciones entre los subsistemas y los componentes del *software* y detallan las reglas para especificar las relaciones entre los elementos (clases, paquetes, subsistemas) de la arquitectura. Para modelar la arquitectura del sistema se utilizó el patrón arquitectónico Programación por Capas, en este caso la arquitectura en dos capas. Esta arquitectura otorga a cada nivel una misión simple, lo que permite el diseño de arquitecturas escalables, o sea, que pueden ampliarse con facilidad en caso de que las necesidades aumenten.

En la arquitectura en dos capas o clientes/servidor, el cliente es quien inicia un requerimiento de servicio y el servidor es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LANs<sup>17</sup> o WANs<sup>18</sup>, para proveer de múltiples servicios a los clientes y ciudadanos tales como impresión, acceso a bases de datos, fax y procesamiento de imágenes.

---

<sup>17</sup> Red de área local. Es la interconexión de una o varias computadoras y periféricos. Su extensión está limitada físicamente a un edificio o a un entorno de 200 metros, con repetidores podría llegar a la distancia de un campo de 1 kilómetro.

<sup>18</sup> Red de área amplia. tipo de red de computadoras capaz de cubrir distancias desde unos 100 hasta unos 1000 km, proveyendo de servicio a un país o un continente.



**Figura 3: Vista del Patrón Arquitectónico Modelo en dos capas**

El patrón arquitectónico Programación por Capas separa la lógica de negocio y la presentación, por lo que se consigue un mantenimiento más sencillo de las aplicaciones. (Potencier Fabien, 2008). Aunque Drupal tiene una arquitectura basada en tres capas, esto no es un impedimento para que los módulos que se realicen utilizando este CMS no implementen esta arquitectura.

### 3.3. Patrones de Diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de *software* y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño es una solución a un problema de diseño. Los patrones de diseño describen una estructura que resuelve un problema de diseño en particular. Se aplican a un elemento específico del diseño, relaciones entre componentes o los mecanismos para efectuar la comunicación de componente a componente. (Pressman, 2005). Los patrones de diseño se clasifican GRASP y GOF.

Los patrones de asignación de responsabilidades GRASP, permiten asignar correctamente responsabilidades a cada una de las clases. Se aplican durante la elaboración de los diagramas de interacción al asignar las responsabilidades a los objetos y al diseñar la colaboración entre ellos. Describen los principios fundamentales de la asignación de responsabilidades a objetos. Los patrones GRASP son: experto, creador, alta cohesión, bajo acoplamiento y controlador. (Larman, 1999). De este grupo de patrones se utilizaron los siguientes:

- ✓ **Controlador:** Sirve de intermediario entre una clase interfaz y la clase que contiene el algoritmo de la funcionalidad. El uso de este patrón se encuentra reflejado en la clase `theme_config.module`, debido a que es la clase que se encarga de gestionar todos los eventos de entrada al módulo.

- ✓ **Experto:** Permite asignar responsabilidades entre las clases, estableciendo para cada clase su responsabilidad de acuerdo a la información que posee. El uso de este patrón se encuentra reflejado en la clase `theme_config.module`, pues es la clase que cuenta con la información necesaria para cumplir las peticiones del administrador.
- ✓ **Bajo acoplamiento:** Define el uso mínimo de clases entrelazadas de modo que si se hacen modificaciones solamente afecten a la menor cantidad de clases. El uso de este patrón se encuentra reflejado en el módulo implementado debido a que el nivel de dependencias entre las clases es muy bajo.
- ✓ **Alta Cohesión:** Permite asignar responsabilidades entre las clases, estableciendo para cada clase su responsabilidad de acuerdo a la información que posee. El uso de este patrón se encuentra reflejado en la manera en que se declararon los métodos, así como la organización de los mismos dándole a cada clase la responsabilidad que le toca.

En el *CMS Drupal* no existen clases declarativas explícitamente. El diseño del mismo hace uso de varios elementos de la Programación Orientada a Objetos (POO) dentro de los que se encuentran: herencia, polimorfismo y encapsulamiento, entre otros elementos y características que lo hacen Orientado a Objetos. Drupal utiliza patrones de diseño característicos de sistemas Orientados a Objetos, particularmente los patrones *GOF*, los cuales se dividen en varias categorías entre las que se encuentran: creacionales, estructurales y de comportamiento. (Martín, 2010). Los patrones que se manifiestan específicamente en el funcionamiento de Drupal son:

- ✓ **Fachada:** Brinda una interfaz que puede conectar o servir de intermediaria con uno o varios sistemas más complejos. Logra tener un acceso más sencillo para que los clientes puedan tener una interacción completa con el sistema. El patrón se encuentra reflejado en la clase `theme_config.module`, debido a que es la encargada de crear los métodos para poder realizar las modificaciones a la plataforma. (Larman, 1999)

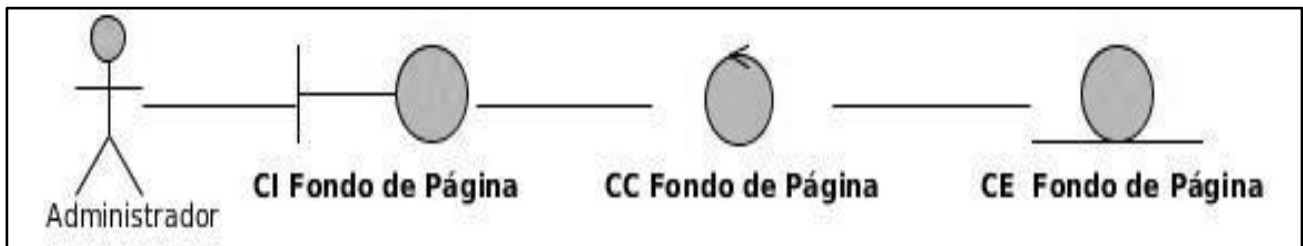
#### 3.4. Diagramas de Clases del Análisis

Describe la estructura del sistema a desarrollar mostrando sus clases, atributos y relaciones entre ellos. Son utilizados durante el proceso de análisis y diseño de los sistemas para crear un diseño conceptual de la información que el mismo manejará y los componentes encargados de su funcionamiento. Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, muestra los bloques de construcción de cualquier sistema orientado a objetos, los diagramas de Clase

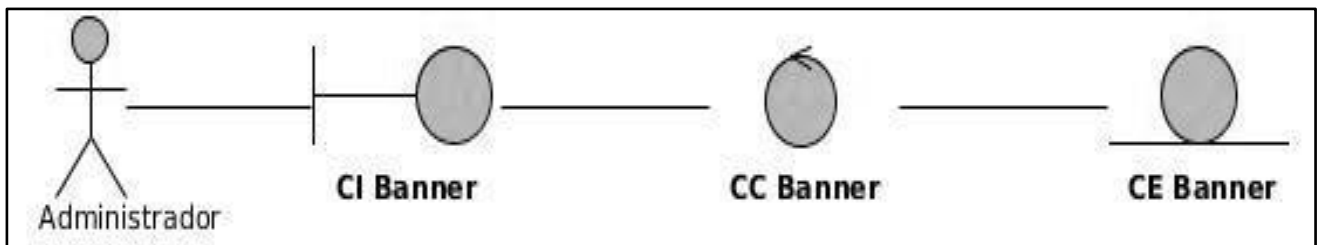
son más útiles para ilustrar relaciones entre clases e interfaces. Un diagrama de clases está compuesto por los siguientes elementos:

**Clase:** atributos, métodos y visibilidad.

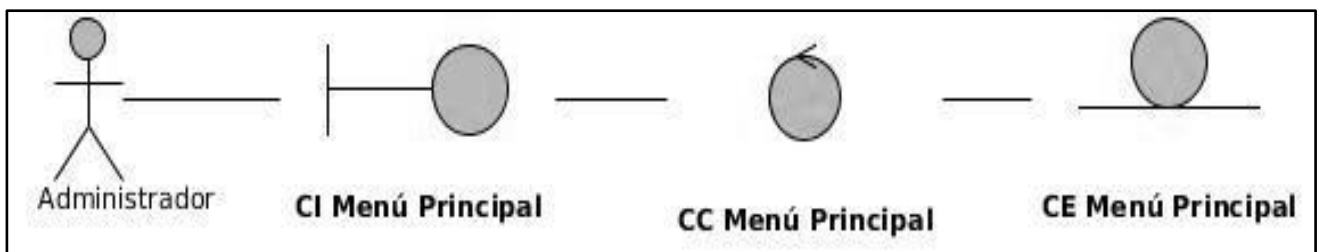
**Relaciones:** Herencia, Composición, Agregación, Asociación y Uso. (Scribd, 2006)



*Figura 4: Diagrama de clases del análisis del CU Cambiar Fondo de Página*



*Figura 5: Diagrama de clases del análisis del CU Modificar Banner*



*Figura 6: Diagrama de clases del análisis del CU Configurar Menú Principal*

### 3.5. Diagrama de Clases del Diseño

Los diagramas de clases del diseño son los diagramas principales en el flujo de trabajo análisis y diseño para obtener un sistema. Durante el diseño del sistema, el diagrama de diseño se desarrolla enfocado a los detalles de las implementaciones. (Larman, 1999). A continuación se muestran los diagramas de clases del diseño según la personalización de los usuarios de la solución:

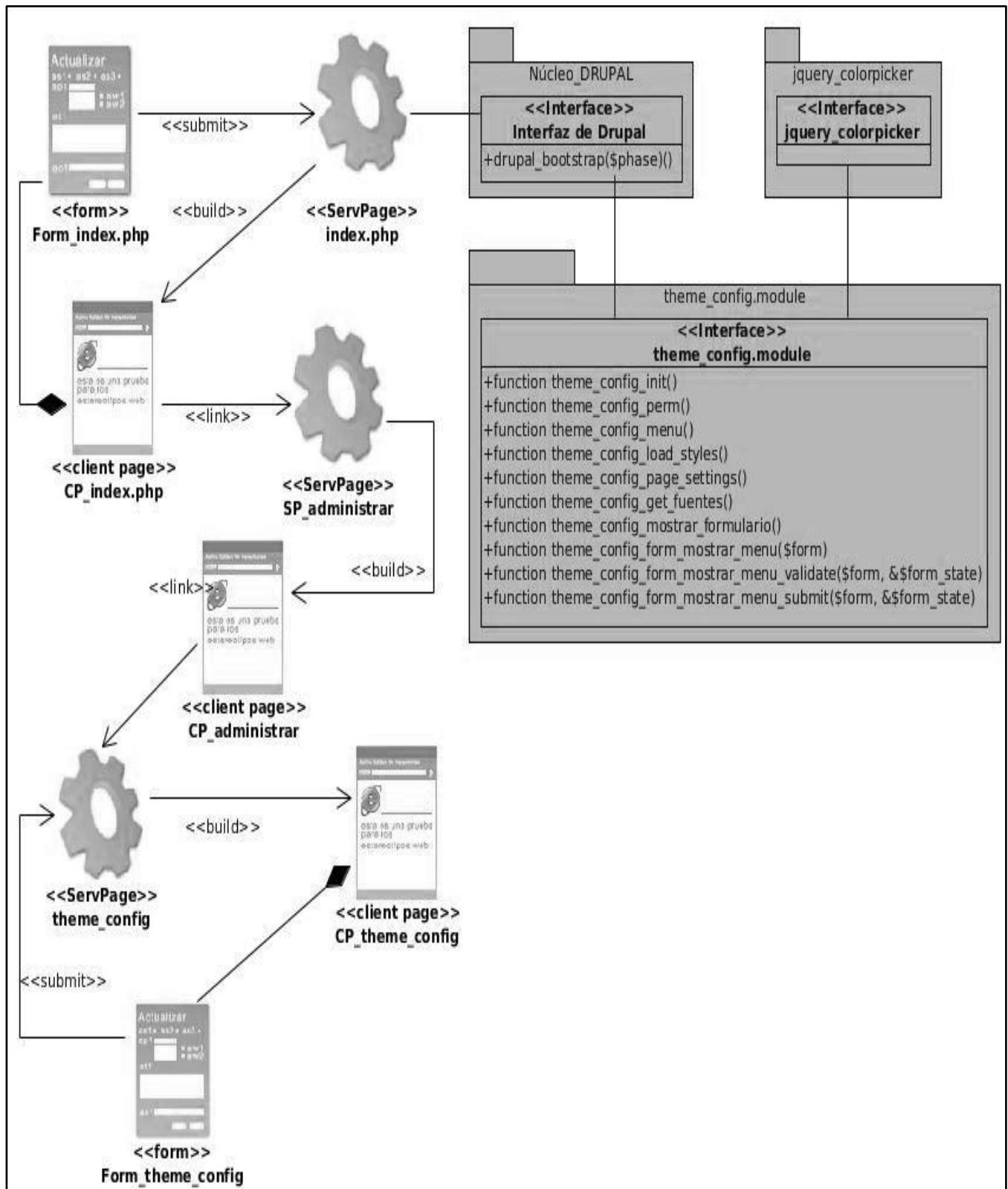


Figura 7: Diagrama de clases del diseño

### **3.6. Diagramas de Interacción**

Los diagramas de interacción muestran un conjunto de objetos y sus relaciones, incluyendo los mensajes que puedan ser realizados entre ellos. Son importantes para modelar los aspectos dinámicos de un sistema y para construir sistemas ejecutables a través de ingeniería hacia adelante e ingeniería inversa. Comúnmente contienen objetos, enlaces y mensajes. Pueden servir para visualizar, especificar, construir y documentar los aspectos dinámicos de una sociedad particular de objetos, o pueden ser usados para modelar un flujo particular de control de un caso de uso. Los diagramas de interacción están conformados por los diagramas de secuencia y los diagramas de colaboración. (Jacobson, y otros, 1999)

#### **3.6.1. Diagramas de Colaboración**

Los diagramas de colaboración (Anexo 1) muestran las interacciones entre objetos creando enlaces entre ellos y añadiendo mensajes a esos enlaces. Se utilizan para ilustrar la realización de un caso de uso, muestra como los objetos interactúan para lograr el comportamiento de un caso de uso o parte de este y de esta forma definen los roles de los mismos. A diferencia de los diagramas de secuencia su principal objetivo es mostrar la relación entre dichos objetos. Estos diagramas tienen una mayor utilidad cuando se utilizan en interacciones entre un número no muy grande de objetos, pues en caso contrario el número de mensajes entre estos crece y el diagrama se hace difícil de entender; en estos casos los diagramas de secuencia son una mejor elección. (Jacobson, y otros, 1999)

#### **3.6.2. Diagramas de Secuencia**

Un diagrama de secuencia explica gráficamente las interacciones existentes entre las instancias de las clases del modelo de estas. El punto de partida de las interacciones es el cumplimiento de las poscondiciones de los contratos de operación, se utilizan para ilustrar la realización de un caso de uso, son particularmente importantes para los diseñadores pues aclaran los roles jugados por los objetos en un flujo, lo cual le proporciona un gran valor para la determinación de las responsabilidades de las clases y además incluye una secuencia cronológica de los mensajes y no la relación entre los objetos, por lo que es mejor su utilización cuando el orden en el tiempo de los mensajes es de importancia. (Jacobson, y otros, 1999)

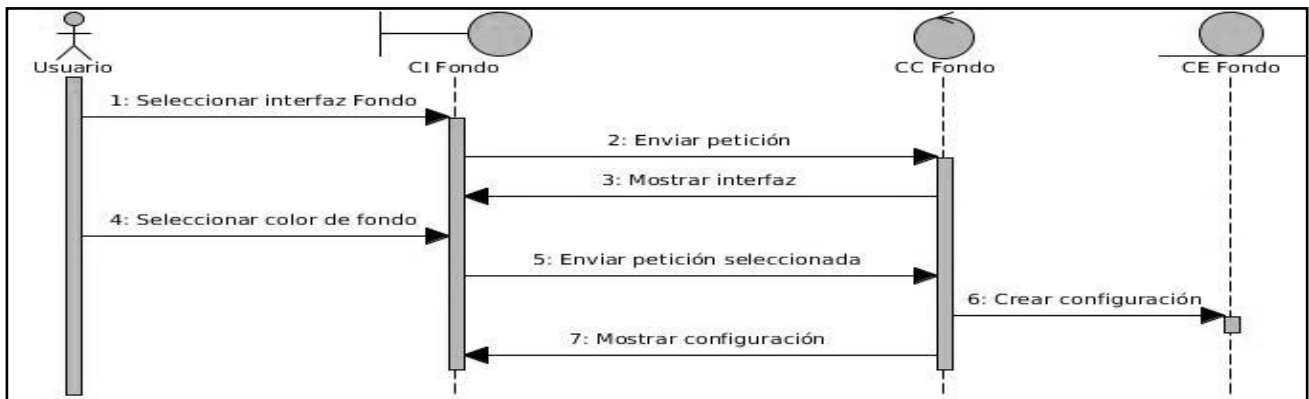


Figura 8: Diagrama de Secuencia del CU Modificar Fondo de Página

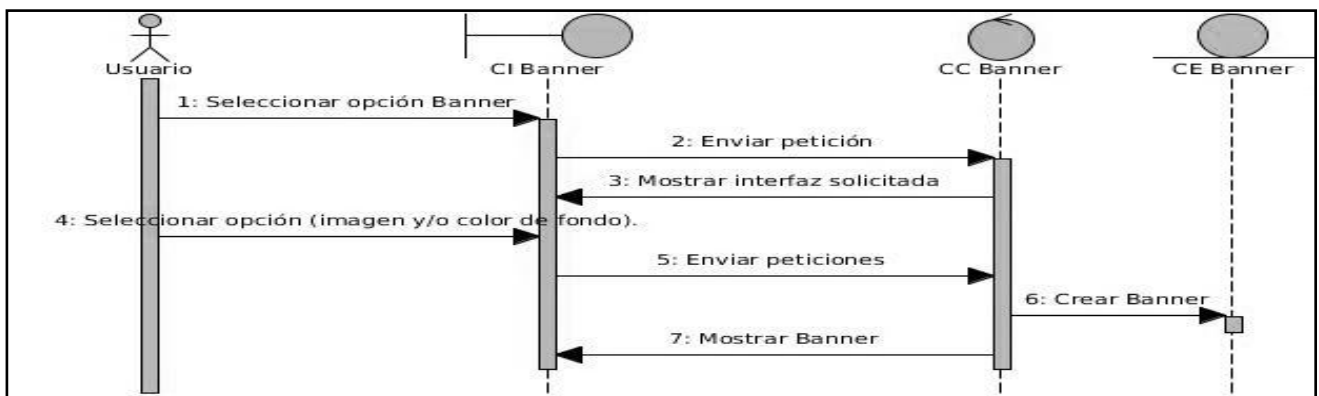


Figura 9: Diagrama de Secuencia del CU Modificar Banner

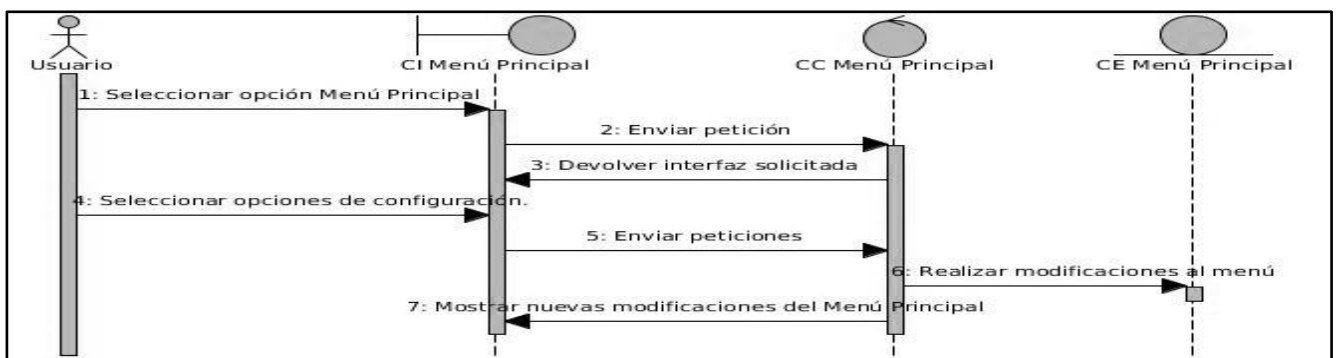


Figura 10: Diagrama de Secuencia del CU Modificar Menú Principal

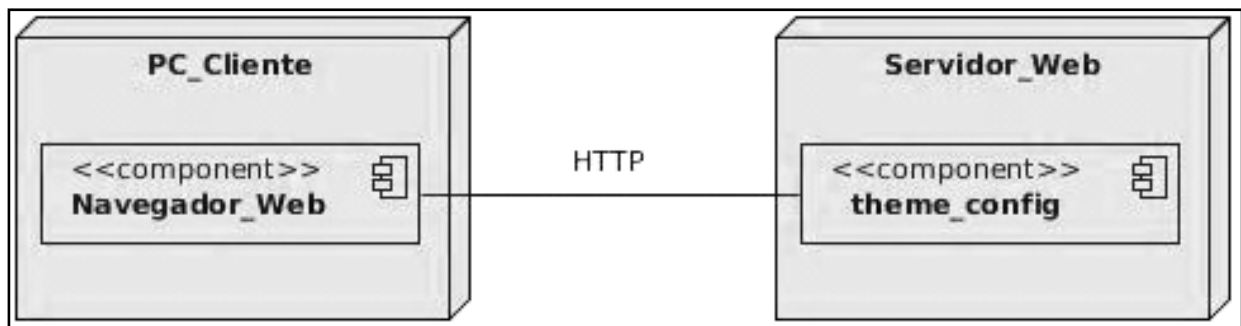
### 3.7. Diagrama de Despliegue

Diagrama del Lenguaje Unificado de Modelado que muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos, la configuración



en funcionamiento del sistema incluyendo su *software* y su *hardware*. Son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. (Jacobson, et al., 1999). Estos diagramas están compuestos por:

- ✓ **Nodos:** Elementos de procesamiento con al menos un procesador, memoria y otros dispositivos.
- ✓ **Conectores:** Expresan el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.
- ✓ **Dispositivos:** Nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.



**Figura 11: Diagrama de Despliegue**

#### Descripción de los nodos:

- ✓ **PC\_Cliente:** Nodo cliente mediante el cual el usuario accederá a la aplicación utilizando un navegador *web*.
- ✓ **Servidor\_Web:** Nodo en el cual se encontrará instalado el servidor *web Apache* que será donde se alojará la aplicación *web* que contiene el componente.

### 3.8. Conclusiones

En este capítulo se realizó un estudio para definir la arquitectura de *software* y los patrones a utilizar en el diseño e implementación de la solución propuesta. La modelación de los diagramas de secuencia permitió lograr un mejor entendimiento de los flujos de los casos de uso. Se hizo una descripción del diagrama de despliegue lo que acarreó una mejor visión de la distribución física del sistema. Se modelaron los diagramas de clases del diseño permitiendo definir los principales métodos y atributos a desarrollar durante el proceso de implementación del módulo. Una vez concluido el flujo de Análisis y Diseño se está en condiciones de comenzar con la implementación del módulo de personalización.



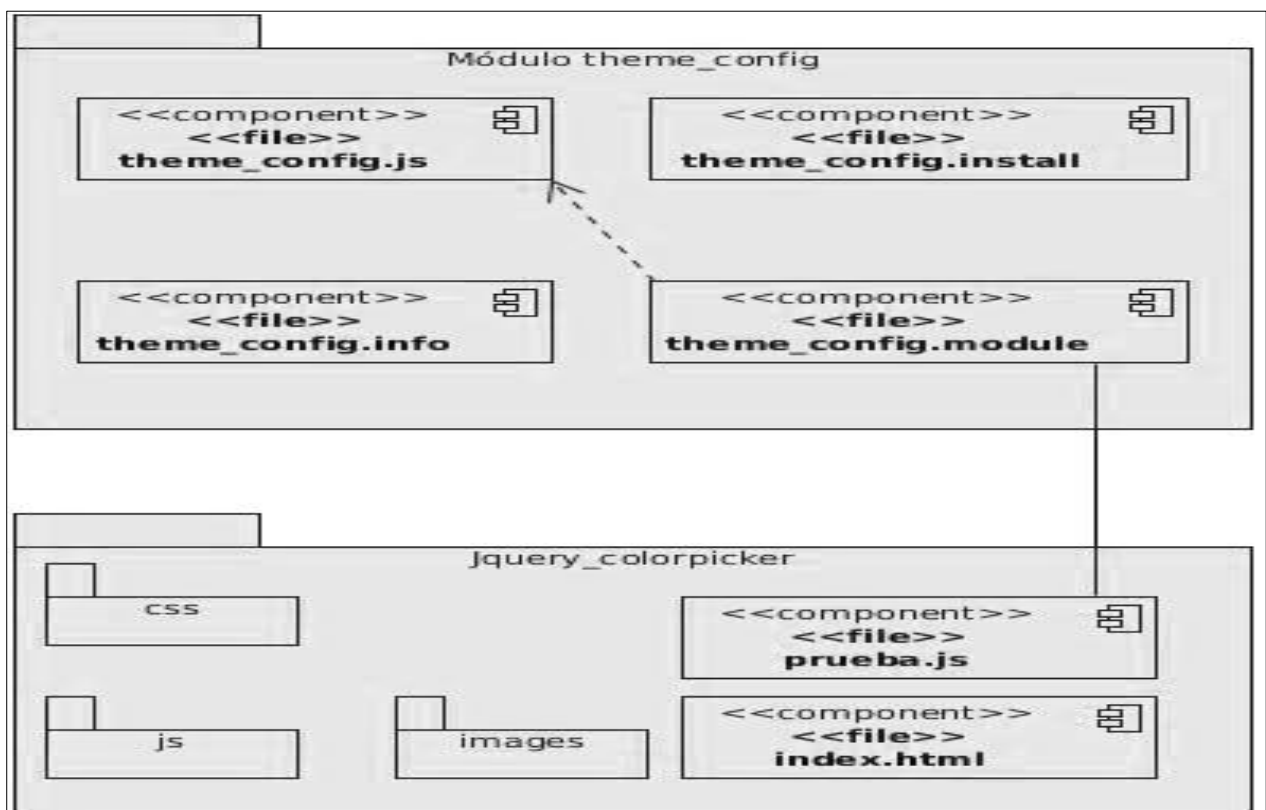
## Capítulo IV: Implementación y Pruebas

### 4.1. Introducción

El capítulo IV trata sobre los flujos de trabajo implementación y prueba, donde se desarrolla el diagrama de componentes que describe los elementos físicos del sistema y sus relaciones. Mediante el diagrama de despliegue se muestran los diferentes nodos que componen el sistema y se aplican las pruebas a la aplicación para comprobar si cumple los requerimientos del cliente.

### 4.2. Diagrama de Componentes

Muestra como el sistema está dividido en componentes y las dependencias entre ellos. Ayuda a los desarrolladores a visualizar el camino de la implementación, permiten modelar la disposición física o topología de un sistema. Se utiliza para modelar la vista estática de un sistema, la organización y las dependencias entre un conjunto de clases. (Jacobson, et al., 1999)



**Figura 12: Diagrama de Componentes**

### 4.3. Estándares de Codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. El estándar en el código de programación es muy importante en cualquier proyecto, principalmente si este involucra muchos desarrolladores, lo que asegura que el código sea de alta calidad, que contenga una cantidad baja de errores y sea fácil de mantener.

#### Criterios de Calidad

Es importante que durante la codificación se consideren permanentemente los siguientes criterios de calidad:

Criterio	Objetivo
Facilidad de Comunicación	Proporcionar al usuario entradas y salidas fácilmente asimilables.
Auto-descripción	Proporcionar en el código, explicaciones sobre la implantación realizada.
Simplicidad	La implantación realizada debe hacerse de la forma más comprensible posible.

#### Principios Generales

- Los nombres de cada uno de los elementos del programa deben ser significativos; su nombre debe explicar en lo posible el uso del elemento.
- No manejar en los programas más de una instrucción por línea.
- Declarar las variables en líneas separadas.
- Añadir comentarios descriptivos junto a cada declaración de variables, si es necesario.
- La mayoría de los elementos se deben nombrar usando sustantivos.
- Los atributos deben comenzar con letra minúsculas y los métodos deben comenzar con letra mayúsculas.

#### Cada función debe tener un encabezado que contenga:

- Objetivo de la función y no descripción del procedimiento.

- Comentarios de apoyo a variables, llamadas a función o inclusión de archivos que no sean obvios al proceso.
- Explicación de uso de argumentos (parámetros) no obvios.
- Explicación de uso de valores devueltos (de retorno).

### Se utilizará dos tipos de comentarios:

*/\* Se puede utilizar en el caso de un atributo, de un método o algún segmento de código \*/*

### Otros Comentarios Internos:

Los otros comentarios necesarios deben figurar a partir de la columna 35.

```
// .....  
//.....
```

### Aspectos específicos sobre PHP

- ✓ Indentación
  - Define que debe usarse los sangrados para indentar el código y nunca espacios.
- ✓ Saltos de Línea
  - Añadir un salto de línea después del cierre de los paréntesis de los parámetros.
  - Añadir un salto de línea después un punto y coma, cuando termina la sentencia.
- ✓ Espacios y líneas en blanco
  - Usar espacios en blanco para mejorar la legibilidad del código.
  - Usar espacios en blanco en ambos lados del operador de símbolos, después de comas y después de las declaraciones.
  - Usar líneas en blanco para separar trozos de código de alta complejidad.
- ✓ Longitud de la línea
  - Evitar las líneas de más de 80 caracteres cuando se supere esta cifra se debe reorganizar el código usando algún principio descrito anteriormente.

### Otros aspectos sobre PHP

- ✓ String Literales
  - Cuando se le asigna un texto literal (sin contenido de variables) se utilizarán comillas simples.  
\$a = 'Texto de ejemplo';

✓ **String Literales con apóstrofes**

- Cuando se le asigna un texto literal con apóstrofes se utilizarán comillas dobles, este caso es principalmente en las sentencias SQL.

```
$sql = "SELECT `id`, `name` from `some_people` WHERE `name`='Fred' OR `name`='Susan';"
```

✓ **Sustitución por Variables**

- Cuando un texto contiene porciones que serán sustituidas por el valor de alguna variable, se toma como correcto únicamente estas dos formas:

```
$greeting = "Hola $nombre, bienvenido!";
```

```
$greeting = "Hola {$nombre}, bienvenido!";
```

✓ **Concatenación**

- Para concatenar Strings se utilizará el operador "." (punto), con un espacio entre medio para mejorar la lectura.

```
$company = 'Zend' . ' ' . 'Technologies';
```

✓ **Arrays**

- En aquellos arrays de índices numéricos estos deberán ser números positivos, en la declaración de los valores del array se dejará un espacio en blanco luego de la coma para mejorar la lectura.

✓ **Clases**

- En la declaración de la clase el nombre deberá seguir la convención ya expresada anteriormente, debe estar precedida por una documentación, luego la declaración y desarrollo.

```
/**
```

```
 * Documentación del bloque aquí.
```

```
*/
```

```
class Tipologia_Controlador {
```

```
 // Todo el contenido de la clase aquí
```

```
 // indentada por cuatro espacios vacíos
```

```
}
```

✓ **Control de flujo**

- En las declaraciones if/then/else deberá tener un espacio antes y después del paréntesis condicional, en la misma línea se abre llave y se cierra en una línea diferente, lo mismo se aplica al elseif, a continuación un ejemplo que lo ilustra:

```
if (condition1 || condition2) {  
    action1;  
}  
elseif (condition3 && condition4) {  
    action2;  
}  
else {  
    defaultaction;  
}
```

#### 4.4. Pruebas de Software

Las pruebas de *software* son un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática. Son procesos que permiten verificar y revelar la calidad de un producto *software*. El objetivo de estas pruebas es probar el funcionamiento del *software* en su máxima capacidad y tratar de corregir todos los errores hasta que el producto quede lo más limpio posible. Las mismas son realizadas por el jefe de proyecto, los ingenieros de *software* o los especialistas en pruebas.

Uno de los elementos principales para certificar la calidad de una aplicación informática lo constituye el resultado de las pruebas que se le practican. A un *software* se le pueden realizar disímiles pruebas debido a la gran variedad de tipologías de pruebas que existen, cada una de ellas con metas y estrategias bien definidas. Un control y una gestión de calidad bien implementados, especialmente durante el proceso de desarrollo, aumentan la calidad del sistema final, reducen los costos de desarrollo y acortan el tiempo necesario para el mismo. Para determinar el nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema. (Pressman, 2005). En el presente epígrafe se detallarán las pruebas funcionales y de fiabilidad cometidas al módulo de personalización de la Plataforma VideoWeb.

#### 4.4.1. Pruebas de Caja Negra

Las pruebas de caja negra o pruebas de comportamiento son las que se aplican a la interfaz del *software*. Estas pruebas se concentran en los requisitos funcionales del *software*, permiten al ingeniero de *software* derivar un conjunto de condiciones de entrada que probarán por completo todos los requisitos funcionales del programa. Las mismas tratan de encontrar errores de interfaz, funciones incorrectas o faltantes, errores en estructuras de datos o en acceso a bases de datos externas, errores de comportamiento o diseño y errores de inicialización y término. (Pressman, 2005)

Las pruebas son el último paso para la evaluación de la calidad y el descubrimiento de errores pero no deben considerarse como una red de seguridad. La calidad se incorpora al *software* en todo el proceso de ingeniería, la aplicación apropiada de métodos y herramientas, las revisiones técnicas formales y efectivas junto con la administración y una medición sólida aportan la calidad que se confirma durante las pruebas. Se aplican durante las últimas etapas del proceso de prueba y como resultado de las mismas se derivan un conjunto de casos de prueba donde se recoge el comportamiento del sistema.

Las pruebas se llevan a cabo sobre la interfaz del *software* obviando el comportamiento interno y la estructura del programa. Un caso de prueba es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular. Los casos de pruebas se pueden derivar de los casos de uso del sistema o de la realización de estos en el modelo de diseño, permitiendo así validar los requisitos funcionales del sistema. (Pressman, 2005)

Para la ejecución de las pruebas a la solución se desarrollaron seis casos de pruebas correspondientes a los casos de uso del sistema. Los casos de pruebas desarrollados son los siguientes:

##### **CP: Modificar Fondo Página**

**Descripción General:** El caso de uso se inicia cuando el administrador desea cambiar el fondo que tiene la página y selecciona la opción Fondo, finalizando el mismo, cuando el fondo de la página es cambiado.

**Precondiciones:** El administrador debe estar autenticado.

### Secciones a probar en el Caso de Uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Modificar Fondo Página.	EC 1.1: Modificar Fondo Página con éxito.	El administrador selecciona la opción "Fondo". El sistema muestra las opciones que tiene el administrador para modificar el fondo, dándole la posibilidad al usuario de seleccionar un color de relleno o una imagen para aplicar al fondo de la página. El administrador selecciona el color o la imagen que desea aplicar para cambiar el Fondo de la Página y el sistema visualiza los cambios.	Autenticarse/ Administración/ Configuración de los Temas.
	EC 1.2: Modificar Fondo Página sin éxito.	El administrador no selecciona ningún color ni imagen y el sistema no realiza ninguna transformación.	Autenticarse/ Administración/ Configuración de los Temas.

### Descripción de las variables

No	Nombre de	Clasificación	Valor Nulo	Descripción
1	Color de Fondo	Lista desplegable	Sí	Solo admite letras.

### Secciones a revisar

Id del escenario	1	Respuesta del sistema	Resultado de la prueba
EC1.1: Modificar Fondo de la Página con éxito.	V/ "Amarillo"	El sistema muestra las opciones que tiene el administrador para modificar el fondo, dándole la posibilidad al usuario de seleccionar un color de relleno o una imagen para aplicar al fondo de la página. El sistema visualiza los cambios.	Satisfactoria
EC1.1: Modificar Fondo de la Página sin éxito.	I/"Nulo"	El sistema no realiza ninguna transformación.	Satisfactoria

### CP: Modificar Banner

**Descripción General:** El caso de uso se inicia cuando el administrador desea modificar el encabezado o *banner* de la página y selecciona la opción *Banner*, finalizando el mismo cuando el administrador obtiene las modificaciones deseadas.

**Precondiciones:** El administrador debe estar autenticado.

#### Secciones a probar en el Caso de Uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Modificar <i>Banner</i> .	EC 1.1: Modificar <i>Banner</i> con éxito.	El administrador selecciona la opción " <i>Banner</i> ". El sistema brinda la posibilidad de seleccionar un color para cambiar el color de fondo de la página. El administrador selecciona el color que desea tenga el <i>banner</i> de la página y el sistema visualiza en la página el nuevo color del banner.	Autenticarse/ Administración/ Configuración de los Temas.
	EC 1.2: Modificar <i>Banner</i> sin éxito.	El administrador no selecciona ningún color para aplicar al <i>banner</i> de la página y el sistema no realiza ninguna transformación.	Autenticarse/ Administración/ Configuración de los Temas.

#### Descripción de las variables

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Color de Fondo	Lista desplegable	Sí	Solo admite letras.



### Secciones a revisar

Id del escenario	1	Respuesta del sistema	Resultado de la prueba
EC1.1: Modificar Banner con éxito.	V/ "Verde"	El administrador selecciona la opción "Banner". El sistema brinda la posibilidad de seleccionar un color para cambiar el color de fondo de la página. El administrador selecciona el color que desea tenga el <i>banner</i> de la página y el sistema visualiza en la página el nuevo color del banner.	Satisfactoria
EC1.1: Modificar Banner sin éxito.	I/ "Nulo"	El administrador no selecciona ningún color para aplicar al <i>banner</i> de la página y el sistema no realiza ninguna transformación.	Satisfactoria

### CP: Modificar Menú Principal

**Descripción General:** El caso de uso se inicia cuando el administrador desea configurar el menú principal de la página y selecciona la opción Menú Principal, finalizando el mismo cuando el menú queda configurado.

**Precondiciones:** El administrador debe estar autenticado.

### Secciones a probar en el Caso de Uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Modificar Menú Principal.	EC 1.1: Modificar Menú Principal con éxito.	El administrador selecciona la opción "Menú Principal". El sistema muestra unas listas desplegables para que el administrador seleccione el color, la familia de fuente y el tamaño que desea aplicar al menú principal de la página. El administrador selecciona las transformaciones que desea aplicar al Menú Principal y el sistema procesa la información a partir de la acción realizada y actualiza la visualización en el menú principal.	Autenticarse/ Administración/ Configuración de los Temas.

	EC 1.2: Modificar Menú Principal sin éxito.	El administrador no selecciona ninguna de las opciones de transformación para aplicar al Menú Principal de las que el sistema brinda y el sistema no visualiza ninguna transformación.	Autenticarse/ Administración/ Configuración de los Temas.
--	---	--	---

### Descripción de las variables

No.	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Fuente	Lista desplegable	Sí	Solo admite letras.
2	Tamaño	Lista desplegable	Sí	Solo admite números.
3	Color	Lista desplegable	Sí	Solo admite letras.

### Secciones a revisar

Id del escenario	1	2	3	Respuesta del sistema	Resultado de la prueba
EC1.1: Modificar Menú Principal con éxito.	V/ "Arial"	V/ "30"	V/ "Verde"	El sistema muestra unas listas desplegables para que el administrador seleccione el color, la familia de fuente y el tamaño que desea aplicar al menú principal de la página. El sistema procesa la información a partir de la acción realizada por el administrador y actualiza la visualización en el menú principal.	Satisfactoria
EC1.1: Modificar Menú Principal sin éxito.	I/ "Nulo"	I/ "Nulo"	I/ "Nulo"	El sistema no visualiza ninguna transformación.	Satisfactoria

#### **4.5. Conclusiones**

Como resultado de este capítulo se obtuvo el diagrama de componentes en el cual se muestra la estructura física que tiene la aplicación y las dependencias que hay entre cada uno de los componentes que integran la aplicación. El conocer que tipos de pruebas pueden aplicársele a un sistema informático permitió seleccionar las que se le aplicaron a la solución propuesta. A través de las pruebas seleccionadas se pudo validar el cumplimiento de los requisitos funcionales establecidos en la fase inicial del proceso de desarrollo del producto, constatando el correcto funcionamiento de la Plataforma VideoWeb.

## Conclusiones Generales

Durante el desarrollo del trabajo de diploma y el tiempo dedicado a la investigación, implementación y prueba de la solución propuesta se adquirieron conocimientos necesarios para el desarrollo del sistema y fundamentales para la formación profesional del autor. La culminación de la investigación arrojó los siguientes resultados que dan cumplimiento a los objetivos propuestos:

- ✓ Se realizó un estudio detallado de las definiciones de los conceptos básicos relacionados con el dominio del problema contribuyendo al enriquecimiento del desarrollo de la investigación y logrando un mayor entendimiento de esta.
- ✓ Se caracterizaron varias de las soluciones similares existentes en el mundo y que se utilizan para personalizar aplicaciones *web* y se llegó a la conclusión de que no pueden ser utilizadas porque son propietarias y las que se requieren deben ser libres y multiplataforma.
- ✓ Se analizaron además los lenguajes de programación que se utilizarán para desarrollar la aplicación.
- ✓ Se seleccionaron las herramientas para el funcionamiento adecuado de la aplicación, se decidió utilizar RUP por las facilidades que brinda y la excelente organización en la documentación del trabajo.
- ✓ Se escogió para el desarrollo de la aplicación el *CMS Drupal*, el cual se destaca por su flexibilidad y potencia, contando con módulos libres para aumentar sus funcionalidades y la experiencia en la UCI en el desarrollo de aplicaciones informáticas empleando el CMS Drupal.
- ✓ El patrón arquitectónico Modelo Vista Controlador (MVC) se seleccionó para el desarrollo de los módulos necesarios para la Plataforma VideoWeb, ya que facilita el trabajo en equipo y permite la flexibilidad de los componentes desarrollados.
- ✓ Se plantearon los requisitos no funcionales que influyen de manera significativa en la Arquitectura de *Software*. Se definieron las vistas arquitectónicas (Vista de casos de uso, Vista

Lógica, Vista de procesos, Vista de despliegue y Vista de Implementación) vitales para el desarrollo de la plataforma, describiendo los principales componentes y su interrelación.

- ✓ La solución final tendrá un gran impacto al estar compuesta en su totalidad por tecnologías de *software* libre, con gran soporte por comunidades tanto internacionales como dentro de la Universidad, lo cual contribuye considerablemente al prestigio y la calidad de la misma y a la migración a *software* libre de Inter-nos.
  
- ✓ La arquitectura permitirá que el producto sea modificable, flexible y reusable adaptándose fácilmente a varias distribuciones físicas en dependencia de las necesidades de los clientes, los procesos de la plataforma están pensados para agilizar y facilitar la gestión de los contenidos.

Se confirma haber dado cumplimiento al objetivo general trazado al inicio del trabajo pues se logró terminar el módulo de personalización para la interfaz gráfica de la Plataforma VideoWeb garantizando mayor rapidez y mejor funcionamiento a la hora de realizar cualquier modificación o cambio en la Plataforma.

## Recomendaciones

Al finalizar el presente trabajo se recomienda:

1. Dar continuidad al desarrollo y soporte del módulo desarrollado.
2. Agregar nuevas funcionalidades al módulo de manera que permita mejorar la configuración de la interfaz gráfica de la Plataforma VideoWeb.

## Referencias Bibliográficas

1. **Alvarez, Miguel Angel. 2003.** desarrolloweb.com. *Web Easy Profesional*. [En línea] 2003. [Citado el: 15 de Octubre de 2011.]  
<http://www.desarrolloweb.com/articulos/web-easy-profesional.html>.
2. **Bru, Jose Luis. 2012.** tinkle. *tinkle*. [En línea] 23 de Marzo de 2012. [Citado el: 30 de Marzo de 2012.] <http://www.tinkle.es/blog/tag/personalizacion-de-contenidos>.
3. **Cáceres Tello, Jesús. 2009.** Universidad de Alcalá. *Diagrama de Casos De Uso*. [En línea] 2009. [Citado el: 15 de Marzo de 2012.]  
<http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>.
4. **Definicion.de. 2003.** Definicion.de. *Definicion.de*. [En línea] WordPress, 2003. [Citado el: 5 de Noviembre de 2011.] <http://definicion.de/aplicacion>.
5. —. **2003.** Definicion.de. *Definicion.de*. [En línea] WordPress, 2003. [Citado el: 5 de Noviembre de 2011.] <http://definicion.de/web>.
6. —. **2003.** Definicion.de. *Definicion.de*. [En línea] 2003. [Citado el: 5 de Noviembre de 2011.] <http://definicion.de/pagina-web>.
7. —. **2003.** Definicion.de. *Definicion.de*. [En línea] 2003. [Citado el: 5 de Noviembre de 2011.] <http://definicion.de/sitio-web>.
8. **Djthreem. 2011.** Intercambio Warez. *Microsoft Expression*. [En línea] 30 de Marzo de 2011. [Citado el: 18 de Octubre de 2011.]  
<http://www.intercambiowarez.org/software/microsoft-expression-studio-4-ultimate-espanol-crea-paginas-web-dinamicas-y-de-alta-calidad>.
9. **Domínguez, Alejandro. 2005.** Ayuda Joomla. *Ayuda Joomla*. [En línea] 25 de Mayo de 2005. [Citado el: 20 de Octubre de 2011.]  
<http://ayudajoomla.com/disenio/general/46-artisteer-generador-de-plantillas-para-joomla.html>.
10. **EcuRed. 2010.** EcuRed. *Modelo de Dominio*. [En línea] 14 de Diciembre de 2010. [Citado el: 15 de Febrero de 2012.]  
[http://www.ecured.cu/index.php/Modelo\\_de\\_dominio](http://www.ecured.cu/index.php/Modelo_de_dominio).
11. —. **2010.** EcuRed. *Artisteer*. [En línea] 14 de Diciembre de 2010. [Citado el: 14 de Noviembre de 2011.] <http://www.ecured.cu/index.php/Artisteer>.
12. —. **2010.** EcuRed. *Lenguaje de Programación*. [En línea] 14 de Diciembre de 2010. [Citado el: 10 de Enero de 2012.]  
[http://www.ecured.cu/index.php/Lenguaje\\_de\\_Programaci%C3%B3n](http://www.ecured.cu/index.php/Lenguaje_de_Programaci%C3%B3n).
13. —. **2010.** EcuRed. *Requisitos no Funcionales*. [En línea] 14 de Diciembre de 2010. [Citado el: 18 de Febrero de 2012.]

14. **IWorld.com. 2006.** IWorld.com. *IWorld.com*. [En línea] 2006. [Citado el: 20 de Octubre de 2011.]  
<http://www.adm3.unlp.edu.ar/regular/material/tecnicasdepersonalizacion.pdf>.
15. **Latina Osmosis, Sitio. 2000.** Osmosis Latina. *Guía de UML*. [En línea] 2000. [Citado el: 20 de Octubre de 2011.] <http://www.osmosislatina.com/lenguajes/uml/basico.htm>.
16. **Linares Armas, Dayamy. 2008.** eumed.net. *Requisitos Funcionales*. [En línea] 2008. [Citado el: 12 de Marzo de 2012.]  
<http://www.eumed.net/libros/2010b/698/Requisitos%20funcionales.htm>.
17. **Marin Arias, Marvin David. 2008.** Editboard.com. *Definición de lenguaje de programación. Tipos. Ejemplos*. [En línea] 16 de Octubre de 2008. [Citado el: 25 de Octubre de 2011.]  
<http://catedraprogramacion.foroactivo.net/t83-definicion-de-lenguaje-de-programacion-tipos-ejemplos>.
18. **Marin, Marvin David Arias.** Editboard.com. *Definición de lenguaje de programación. Tipos. Ejemplos*. [En línea] <http://catedraprogramacion.foroactivo.net/t83-definicion-de-lenguaje-de-programacion-tipos-ejemplos>.
19. **NetBeans Team. 2011.** NetBeans. [En línea] 2011. [Citado el: 5 de Diciembre de 2011.]  
<http://netbeans.org/community/releases/roadmap.html>.
20. **Paredes García, Fernando. 2010.** Slideshare. *Drupal, más que un CMS*. [En línea] ISUMMITOXA, 2010. [Citado el: 15 de Enero de 2012.]  
<http://www.slideshare.net/informaticohugo/drupal-como-cms>.
21. **PHP. 2001.** PHP. *PHP Manual*. [En línea] 8 de Noviembre de 2001. [Citado el: 4 de Diciembre de 2011.] <http://www.php.net/manual/es/intro-whatcando.php>.
22. **Politécnica de Catalunya, Universidad. 2002.** UPCommons. *Lenguajes de Programación*. [En línea] 2002. [Citado el: 10 de Diciembre de 2011.]  
<http://upcommons.upc.edu/pfc/bitstream/2099.1/5837/5/part%204.pdf>.
23. **RAE. REAL ACADEMIA ESPAÑOLA. REAL ACADEMIA ESPAÑOLA.** [En línea] [Citado el: 5 de Noviembre de 2011.] <http://buscon.rae.es/drael>.
24. **Sacristán, Luis. 2006.** Sentido Web. *Sentido Web*. [En línea] 6 de Marzo de 2006. [Citado el: 15 de Octubre de 2011.] <http://sentidoweb.com/2008/02/15/guia-para-que-el-usuario-pueda-personalizar-una-web.php>.
25. **Scribd. 2006.** Scribd. *Diagrama de Clases*. [En línea] 2006. [Citado el: 20 de Marzo de 2012.]  
<http://es.scribd.com/doc/35528344/Diagrama-de-Clases>.
26. **Valdez Altamirano, Alfonso. 2009.** ubicuos. *Entornos de Desarrollo Integrados*. [En línea] 2009. [Citado el: 20 de Febrero de 2012.]  
<http://ubicuos.com/wp-content/uploads/2009/05/comparativoides.pdf>.



27. **WordPress.com. 2003.** WordPress.com. *WordPress.com*. [En línea] 7 de Mayo de 2003. [Citado el: 15 de Octubre de 2011.]  
<http://jowicar.wordpress.com/2008/08/14/informatica>.

**Bibliografía**

1. **Alvarez, Miguel Angel. 2003.** desarrolloweb.com. *Web Easy Profesional*. [En línea] 2003. [Citado el: 15 de Octubre de 2011.]  
<http://www.desarrolloweb.com/articulos/web-easy-profesional.html>.
2. **Bru, Jose Luis. 2012.** tinkle. *tinkle*. [En línea] 23 de Marzo de 2012. [Citado el: 30 de Marzo de 2012.] <http://www.tinkle.es/blog/tag/personalizacion-de-contenidos>.
3. **Cáceres Tello, Jesús. 2009.** Universidad de Alcalá. *Diagrama de Casos De Uso*. [En línea] 2009. [Citado el: 15 de Marzo de 2012.]  
<http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>.
4. **Cardona, Fernando. 2011.** *Razonamiento sobre Esquemas Conceptuales UML con Operaciones*. 2011.
5. **Carrillo Pérez Isaías, Rodrigo Pérez González, Rodríguez Martín Aureliano David. 2008.** *Metodología de Desarrollo de Software*. 2008.
6. **Definicion.de. 2003.** Definicion.de. *Definicion.de*. [En línea] WordPress, 2003. [Citado el: 5 de Noviembre de 2011.] <http://definicion.de/aplicacion>.
7. —. **2003.** Definicion.de. *Definicion.de*. [En línea] WordPress, 2003. [Citado el: 5 de Noviembre de 2011.] <http://definicion.de/web>.
8. —. **2003.** Definicion.de. *Definicion.de*. [En línea] 2003. [Citado el: 5 de Noviembre de 2011.] <http://definicion.de/pagina-web>.
9. —. **2003.** Definicion.de. *Definicion.de*. [En línea] 2003. [Citado el: 5 de Noviembre de 2011.] <http://definicion.de/sitio-web>.
10. **Djthreem. 2011.** Intercambio Warez. *Microsoft Expression*. [En línea] 30 de Marzo de 2011. [Citado el: 18 de Octubre de 2011.]  
<http://www.intercambiowarez.org/software/microsoft-expression-studio-4-ultimate-espanol-crea-paginas-web-dinamicas-y-de-alta-calidad>.
11. **Domínguez, Alejandro. 2005.** Ayuda Joomla. *Ayuda Joomla*. [En línea] 25 de Mayo de 2005. [Citado el: 20 de Octubre de 2011.]  
<http://ayudajoomla.com/disenio/general/46-artisteer-generador-de-plantillas-para-joomla.html>.
12. **EcuRed. 2010.** EcuRed. *Modelo de Dominio*. [En línea] 14 de Diciembre de 2010. [Citado el: 15 de Febrero de 2012.]  
[http://www.ecured.cu/index.php/Modelo\\_de\\_dominio](http://www.ecured.cu/index.php/Modelo_de_dominio).
13. —. **2010.** EcuRed. *Artisteer*. [En línea] 14 de Diciembre de 2010. [Citado el: 14 de Noviembre de 2011.] <http://www.ecured.cu/index.php/Artisteer>.

14. —. 2010. EcuRed. *Lenguaje de Programación*. [En línea] 14 de Diciembre de 2010. [Citado el: 10 de Enero de 2012.]  
[http://www.ecured.cu/index.php/Lenguaje\\_de\\_Programaci%C3%B3n](http://www.ecured.cu/index.php/Lenguaje_de_Programaci%C3%B3n).
15. —. 2010. EcuRed. *Requisitos no Funcionales*. [En línea] 14 de Diciembre de 2010. [Citado el: 18 de Febrero de 2012.]
16. **Giraldo Zapata, Luis. 2005.** *Herramientas de Desarrollo de Ingeniería de Software*. 2005.
17. **Gómez Gallego, Juan pablo, Galves, Jorge. 2007.** *FUNDAMENTOS DE LA METODOLOGÍA RUP*. 2007.
18. **IWorld.com. 2006.** IWorld.com. *IWorld.com*. [En línea] 2006. [Citado el: 20 de Octubre de 2011.]  
<http://www.adm3.unlp.edu.ar/regular/material/tecnicasdepersonalizacion.pdf>.
19. **Jacobson, Booch, Rumbaugh. 2000.** *El Proceso Unificado de Desarrollo de Software*. Madrid: Addison Wesley, 2000.
20. **Larman, Craig. 1999.** *UML Y PATRONES. Introducción al análisis y diseño orientado a objetos*. México: PRENTICE HALL, 1999.
21. —. 1999. *UML Y PATRONES. Introducción al análisis y diseño orientado a objetos*. México: PRENTICE HALL, 1999. 970-17-0261-1.
22. **Latina Osmosis, Sitio. 2000.** Osmosis Latina. *Guía de UML*. [En línea] 2000. [Citado el: 20 de Octubre de 2011.] <http://www.osmosislatina.com/lenguajes/uml/basico.htm>.
23. **Linares Armas, Dayamy. 2008.** eumed.net. *Requisitos Funcionales*. [En línea] 2008. [Citado el: 12 de Marzo de 2012.]  
<http://www.eumed.net/libros/2010b/698/Requisitos%20funcionales.htm>.
24. **Marin Arias, Marvin David. 2008.** Editboard.com. *Definición de lenguaje de programación. Tipos. Ejemplos*. [En línea] 16 de Octubre de 2008. [Citado el: 25 de Octubre de 2011.]  
<http://catedraprogramacion.foroactivo.net/t83-definicion-de-lenguaje-de-programacion-tipos-ejemplos>.
25. **Marin, Marvin David Arias.** Editboard.com. *Definición de lenguaje de programación. Tipos. Ejemplos*. [En línea] <http://catedraprogramacion.foroactivo.net/t83-definicion-de-lenguaje-de-programacion-tipos-ejemplos>.
26. **NetBeans Team. 2011.** NetBeans. [En línea] 2011. [Citado el: 5 de Diciembre de 2011.]  
<http://netbeans.org/community/releases/roadmap.html>.
27. **Paredes García, Fernando. 2010.** Slideshare. *Drupal, más que un CMS*. [En línea] ISUMMITOXA, 2010. [Citado el: 15 de Enero de 2012.]  
<http://www.slideshare.net/informaticohugo/drupal-como-cms>.

28. **PHP. 2001.** PHP. *PHP Manual*. [En línea] 8 de Noviembre de 2001. [Citado el: 4 de Diciembre de 2011.] <http://www.php.net/manual/es/intro-whatcando.php>.
29. **Politécnica de Catalunya, Universidad. 2002.** UPCommons. *Lenguajes de Programación*. [En línea] 2002. [Citado el: 10 de Diciembre de 2011.] <http://upcommons.upc.edu/pfc/bitstream/2099.1/5837/5/part%204.pdf>.
30. **Potencier Fabien, Zaninotto François. 2008.** *Symfony la guía definitiva*. 2008.
31. **Pressman, Roger. 2005.** *Ingeniería de Software. Un enfoque práctico*. España: Mc Graw Hill, 2005. 9701054733.
32. —. **2005.** *Ingeniería de Software. Un enfoque práctico*. España: Mc Graw Hill, 2005. 9701054733.
33. **RAE. REAL ACADEMIA ESPAÑOLA. REAL ACADEMIA ESPAÑOLA.** [En línea] [Citado el: 5 de Noviembre de 2011.] <http://buscon.rae.es/drae/>.
34. **S.A. Dart, R.J. Ellison, P.H. Feiler, A.N. Habermann. 1987.** Software Development Environments. *IEEE Computer*. 1987, Vol. 20, 11.
35. —. **1987.** Software Development Environments. *IEEE Computer*. 1987, Vol. 20, 11.
36. **Sacristán, Luis. 2006.** Sentido Web. *Sentido Web*. [En línea] 6 de Marzo de 2006. [Citado el: 15 de Octubre de 2011.] <http://sentidoweb.com/2008/02/15/guia-para-que-el-usuario-pueda-personalizar-una-web.php>.
37. **Scribd. 2006.** Scribd. *Diagrama de Clases*. [En línea] 2006. [Citado el: 20 de Marzo de 2012.] <http://es.scribd.com/doc/35528344/Diagrama-de-Clases>.
38. **Torres Rodríguez, Frank. 2009.** *Desarrollo de la Arquitectura de Software del proyecto Plataforma VideoWeb*. La Habana: s.n., 2009.
39. **Valdez Altamirano, Alfonso. 2009.** ubicuos. *Entornos de Desarrollo Integrados*. [En línea] 2009. [Citado el: 20 de Febrero de 2012.] <http://ubicuos.com/wp-content/uploads/2009/05/comparativoides.pdf>.
40. **WordPress.com. 2003.** WordPress.com. *WordPress.com*. [En línea] 7 de Mayo de 2003. [Citado el: 15 de Octubre de 2011.] <http://jowicar.wordpress.com/2008/08/14/informatica>.

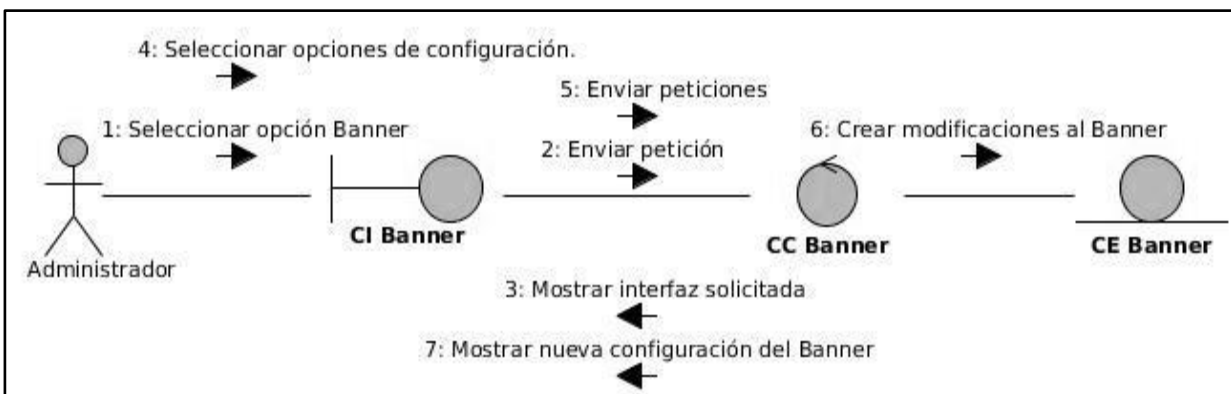
Anexos

Anexo 1: Diagramas de Colaboración

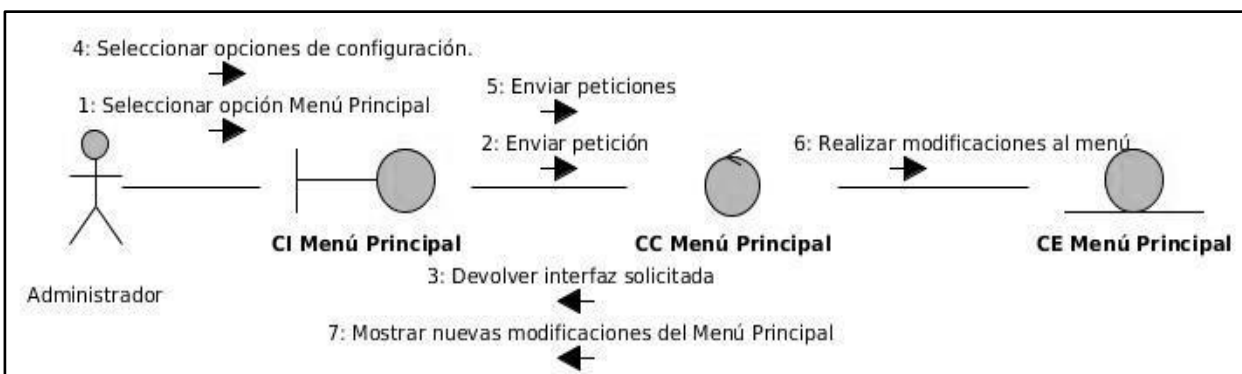
✓ *Diagrama de Colaboración del CU Cambiar Fondo Página*



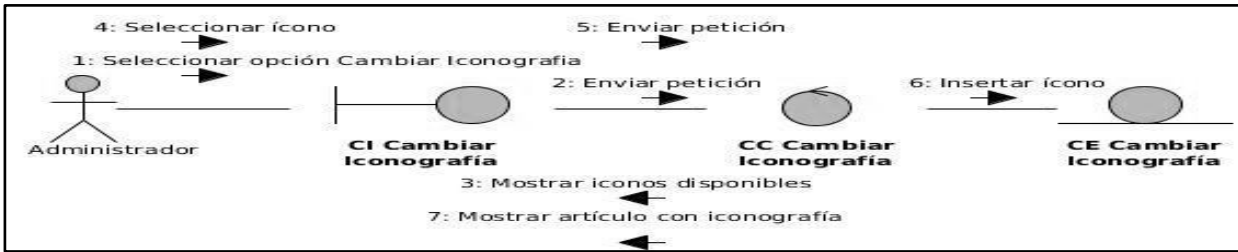
✓ *Diagrama de Colaboración del CU Modificar Banner*



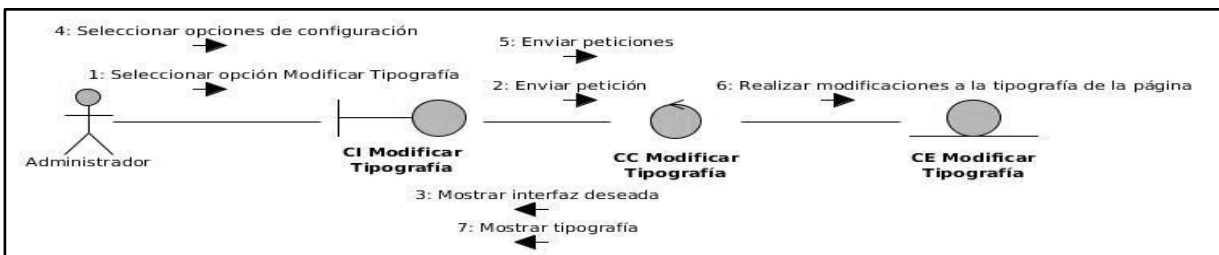
✓ *Diagrama de Colaboración del CU Modificar Menú Principal*



✓ **Diagrama de Colaboración del CU Cambiar Iconografía**



✓ **Diagrama de Colaboración del CU Modificar Tipografía**

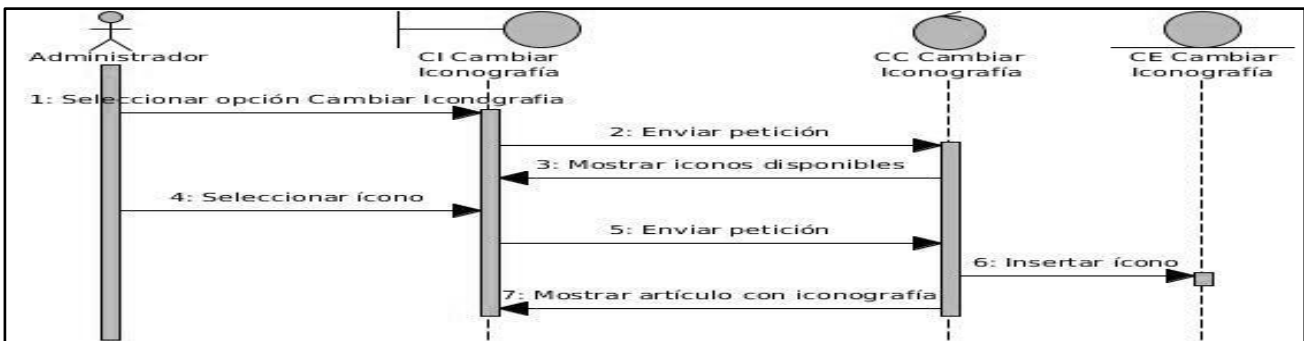


✓ **Diagrama de Colaboración del CU Guardar Cambios**

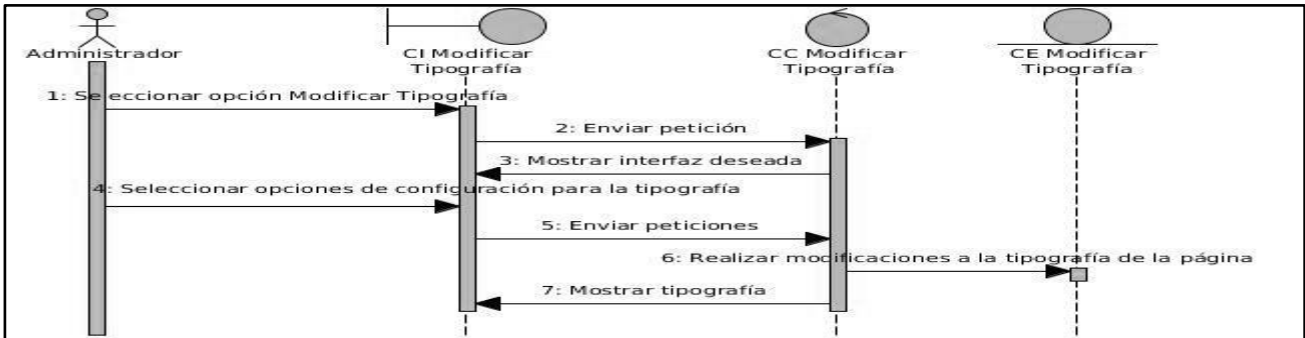


**Anexo 2: Diagramas de Secuencia**

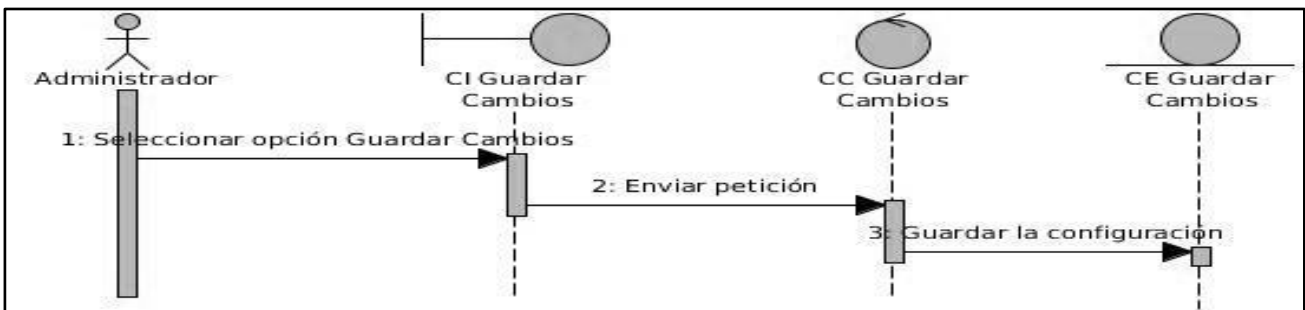
✓ **Diagrama de Secuencia del CU Cambiar Iconografía**



✓ **Diagrama de Secuencia del CU Modificar Tipografía**

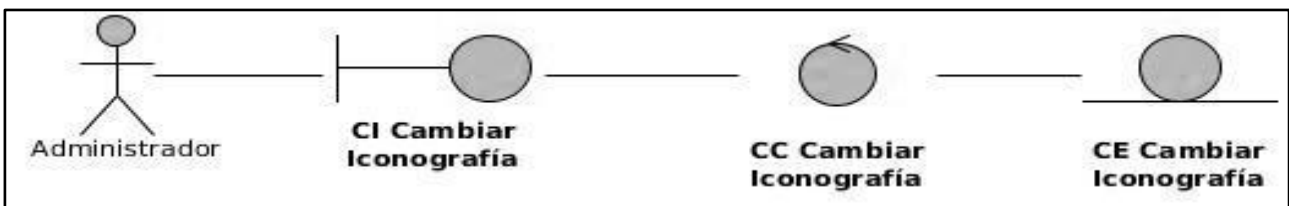


✓ **Diagrama de Secuencia del CU Guardar Cambios**

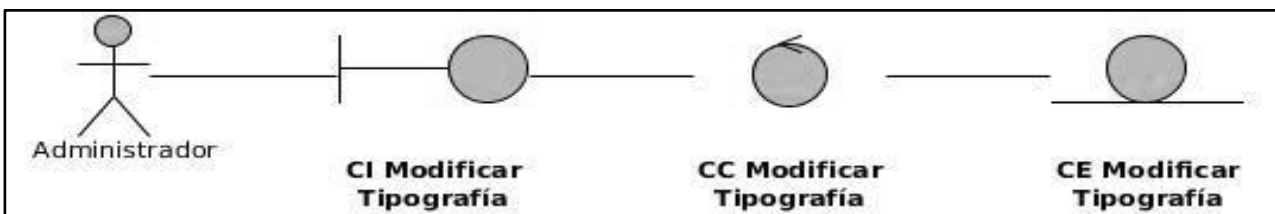


**Anexo 3: Diagramas de Clases del Análisis**

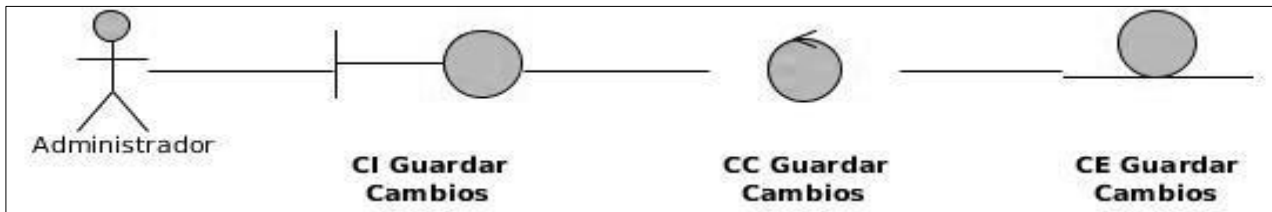
✓ **Diagrama de Clases del Análisis del CU Cambiar Iconografía**



✓ **Diagrama de Secuencia del CU Modificar Tipografía**



✓ **Diagrama de Secuencia del CU Guardar Cambios**



**Anexo 4: Preguntas de la entrevista realizada:**

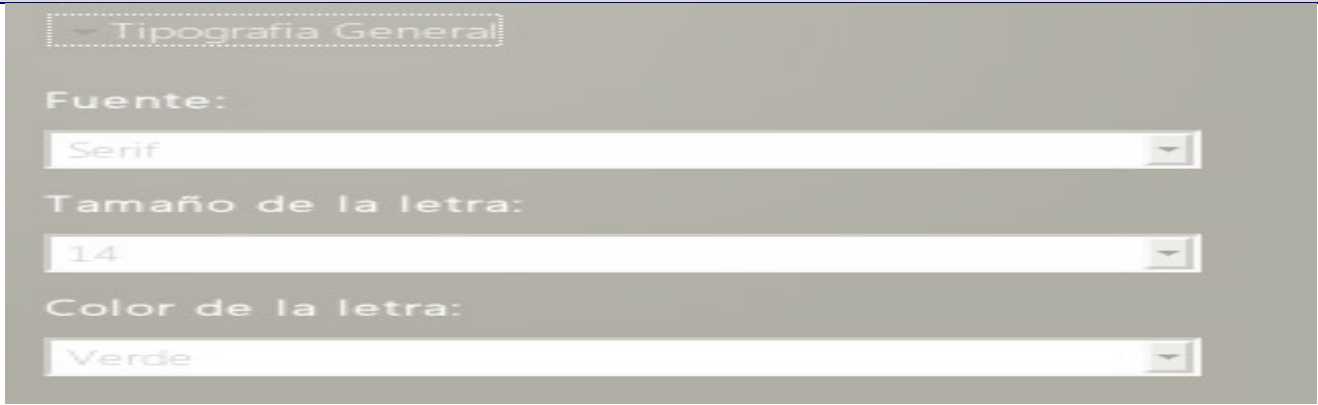
1. ¿Por qué es necesario realizar un módulo para personalizar la interfaz gráfica de la Plataforma VideoWeb?
2. ¿Cómo es que se realiza este proceso actualmente en el proyecto?
3. ¿Qué funcionalidades debe brindar este módulo para satisfacer con las necesidades existentes dentro del proyecto?

**Anexo 5: Descripción de los CU**

✓ **Descripción del CU Modificar Tipografía**

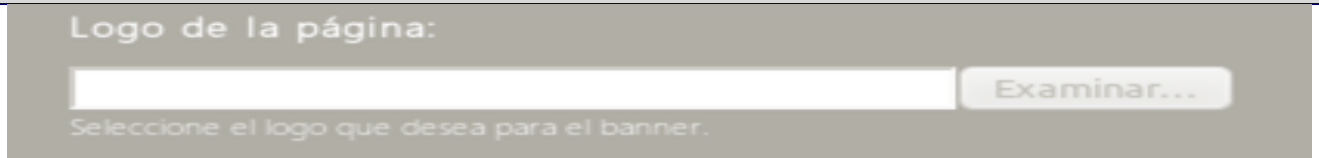
<b>Caso de Uso</b>	Modificar Tipografía
<b>Actores</b>	Administrador
<b>Propósito</b>	El caso de uso tiene el objetivo de brindar al administrador la posibilidad de poder modificar la tipografía que tiene la página.
<b>Resumen</b>	El caso de uso se inicia cuando el administrador desea modificar la tipografía que tiene la página y pulsa la opción Modificar Tipografía.
<b>Precondiciones</b>	El administrador debe estar autenticado.
<b>Referencias</b>	RF8
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador selecciona la opción "Modificar Tipografía".	1.1. El sistema muestra una interfaz donde brinda al administrador una serie de opciones para estructurar el artículo



	<p>de la página, entre ellas se puede encontrar:</p> <ul style="list-style-type: none"> <li>• Formato del texto: El sistema despliega una ventana, para que el administrador seleccione el color, familia de fuente y tamaño que desea tenga la tipografía de la página.</li> </ul>
2. El administrador selecciona los nuevos estilos que tendrá la tipografía de la página.	2.1. El sistema visualiza en la interfaz principal los cambios realizados en la tipografía que se mostrará en la página.
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
2. El usuario no selecciona ningún nuevo estilo para la tipografía de la página.	2.1. El sistema no realiza ninguna transformación.
Prototipo de Interfaz	
	
<b>Poscondiciones</b>	Queda definida la estructura que tendrá la tipografía de la página.


✓ **Descripción del CU Cambiar Iconografía**

<b>Caso de Uso</b>	Cambiar Iconografía
<b>Actores</b>	Administrador
<b>Propósito</b>	El caso de uso tiene como objetivo brindar al administrador la posibilidad de cambiar el ícono existente en el banner de la página.
<b>Resumen</b>	El caso de uso se inicia cuando el administrador desea cambiar el ícono existente en el banner de la página y pulsa la opción "Cambiar iconografía". El caso de uso termina cuando el

	administrador cambia el ícono.
<b>Precondiciones</b>	El administrador debe estar autenticado.
<b>Referencias</b>	RF8
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador desde la interfaz principal selecciona la opción "Cambiar iconografía".	1.1. El sistema muestra una interfaz donde brinda al administrador la posibilidad de cambiar el ícono existente en el banner de la página por otro que él desee.
2. El administrador selecciona el ícono que desea.	2.1. El sistema visualiza en la interfaz principal los cambios realizados en el banner de la página.
<b>Flujo Alternativo</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El usuario no selecciona ninguna imagen como ícono.	2.1. El sistema no realiza ninguna transformación.
<b>Prototipo de Interfaz</b>	
	
<b>Poscondiciones</b>	Queda definido el nuevo ícono que tendrá el banner de la página.

✓ **Descripción del CU Guardar Cambios**

<b>Caso de Uso</b>	Guardar Proyecto
<b>Actores</b>	Usuario
<b>Propósito</b>	Este caso de uso se lleva a cabo con el objetivo de brindar al administrador la posibilidad de guardar una nueva configuración o cualquier cambio realizado sobre a la configuración actual.
<b>Resumen</b>	El caso de uso se inicia cuando el administrador crea una nueva configuración o se realiza algún cambio sobre la configuración existente y pulsa la opción

	"Guardar Cambios".
<b>Precondiciones</b>	El administrador debe estar autenticado.
<b>Referencias</b>	RF7
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador desde la interfaz principal selecciona la opción "Guardar Cambios"	1.1. El sistema guarda la nueva configuración.
<b>Flujo Alterno</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El administrador no selecciona la opción "Guardar Cambios".	2.1. El sistema no guarda la nueva configuración.
<b>Prototipo de Interfaz</b>	
	
<b>Poscondiciones</b>	Quedan guardados los cambios realizados en el proyecto.

## Glosario de Términos

- **Ajax:** Técnica de desarrollo *web* para crear aplicaciones interactivas válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como *JavaScript* y *Document Object Model* (DOM). Tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.
- **DATEC:** Centro de Tecnologías y Gestión de Datos perteneciente a la Facultad 6 de la Universidad de las Ciencias Informáticas. Cuenta con cuatro líneas de desarrollo que apoyan las diferentes áreas productivas de la Universidad y producen *software* de calidad para exportación.
- **Dreamweaver:** Es una aplicación que está destinada a la construcción, diseño y edición de sitios, videos y aplicaciones *web* basados en estándares. Permite al usuario utilizar la mayoría de los navegadores *web* instalados en su ordenador para previsualizar las páginas *web*, dispone de herramientas de administración de sitios dirigidas a principiantes como, por ejemplo, la habilidad de encontrar y reemplazar líneas de texto y código por cualquier tipo de parámetro especificado, hasta el sitio *web* completo.
- **Framework:** Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de *software* concretos, con base a la cual otro proyecto de *software* puede ser más fácilmente organizado y desarrollado. Puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.
- **Geoinformática:** Es el uso de las matemáticas y las técnicas informáticas para resolver problemas geográficos utilizando programas informáticos y modelos matemáticos.
- **GEySED:** Departamento de Geoinformática y Señales Digitales perteneciente a la Facultad 6 de la Universidad de las Ciencias Informáticas, se creó con el objetivo de desarrollar productos

y soluciones informáticas en el área del procesamiento de las Señales Digitales y la Geoinformática.

- **Inter-Nos:** Página totalmente modelada y diseñada para acercar a la población universitaria al conocimiento y la cultura a través del mundo digital.
- **Java:** Lenguaje de programación orientado a objetos, desarrollado por *Sun Microsystems*. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. La memoria es gestionada mediante un recolector de basura.
- **JavaScript:** Lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. *JavaScript* se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo *Java* y *JavaScript* no están relacionados y tienen semánticas y propósitos diferentes. Todos los navegadores modernos interpretan el código *JavaScript* integrado en las páginas *web*. Para interactuar con una página *web* se provee al lenguaje *JavaScript* de una implementación del *Document Object Model* (DOM).
- **Marketing:** Concepto inglés traducido al castellano como mercadeo o mercadotecnia. Se trata de la disciplina dedicada al análisis del comportamiento de los mercados y de los consumidores. Analiza la gestión comercial de las empresas con el objetivo de captar y retener a los clientes a través de la satisfacción de sus necesidades.
- **Patrones GRASP:** Se aplican durante la elaboración de los diagramas de interacción al asignar las responsabilidades a los objetos y al diseñar la colaboración entre ellos. Describen los principios fundamentales de la asignación de responsabilidades a objetos. Los patrones GRASP son: experto, creador, alta cohesión, bajo acoplamiento y controlador.
- **Patrones GOF:** Son los Patrones de Diseño en el campo del Diseño Orientado a Objetos más conocidos y usados en la actualidad.

- **Plugin:** Hace referencia a pequeñas aplicaciones funcionales creadas para ser añadidas a programas más grandes con el propósito de ampliar su capacidad en alguna actividad.
- **PostgreSQL:** Es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES. Es el sistema libre más avanzado con diferencia, soportando la gran mayoría de las transacciones SQL, control concurrente, teniendo a su disposición varios "*language bindings*" como por ejemplo C, C++, Java, Python, PHP y muchos más.
- **Tecnología streaming:** Se utiliza para aligerar la descarga y ejecución de audio y vídeo en la web, permite escuchar y visualizar los archivos mientras se están descargando.