

TRABAJO DE DIPLOMA

MOOX

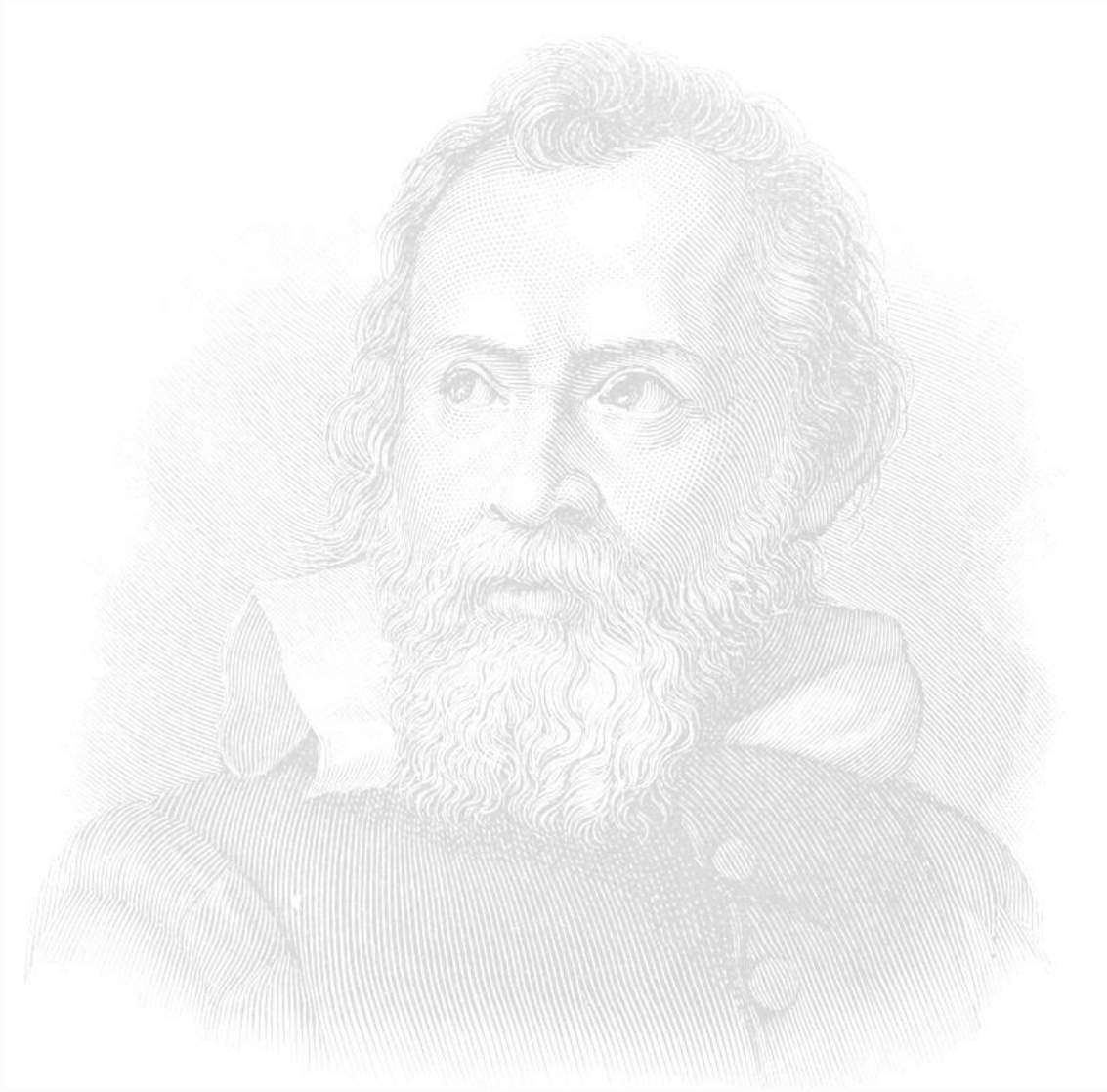
Herramienta generadora de temas visuales
para la plataforma Moodle

Autor:

Jorge Antonio Monedero
Soriano

Tutor:

Ing. Alberto Torres Junco



"Todas las verdades son fáciles de entender, una vez descubiertas. La cuestión es descubrirlas."

Galileo Galilei

Declaración de auditoría

Declaro que soy el único autor de este trabajo y autorizo a 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 _____.
Jorge Antonio Monedero Soriano

Jorge Antonio Monedero Soriano

Ing. Alberto Torres Junco

Firma del autor

Firma del tutor

A quienes me regalaron la vida, y además, me enseñaron a querer vivirla con todas mis fuerzas.

A los que me han dado siempre su apoyo incondicional.

Al dúo que más quiero.

A mis padres.

Doy gracias a mi papa, por brindarme siempre su apoyo, por su amor incondicional y por ser ese padre que me hace sentir dichoso de ser su hijo. A mi mama, la musa de mis ojos, la que ha luchado a mi lado desde siempre y que solo con una mirada es capaz de leer mi mente. No tengo palabras que describan lo agradecido que estoy, a ustedes, que hicieron suyo mi sueño y me brindaron su guía en todo momento, aun cuando no sabía cómo seguir adelante. A mi pequeño hermanito no tan pequeño, esa bestiecilla que entró en mi vida con el objetivo de sacarme de mis cabales, pero a la que amo con todas mis fuerzas. A mi familia, mis tías y tíos, mis primos, mi madrinita. Mis abuelos, a Mama y a Cari que hoy descansan en paz pero este logro es igualmente suyo, a Zule y a Juan, gracias por haberme dado la más feliz de las infancias.

Agradezco a Gustavo, José, Yaiselín, Hugo, Ricardo y Julio por la ayuda que me brindaron en la creación de este trabajo de diploma. A mi guajirita Yise, que ha estado con nosotros desde el origen de los tiempos como un hermanito más. A todos mis compañeros de aula, esos que pertenecieron al grupo 4, el mejor grupo, y a los del grupo 3 que también los quiero muchísimo. A todos mis amigos, con los que he compartido momentos inolvidables, esos campismos a condiciones extremas, esas fiestas. No se quedan mis compañeros del IPI, Dionne, Adonis, Yiliem, Andrés, Ernesto, Bayron, que me enseñaron lo que era la verdadera amistad.

A mi tutor por su apoyo y guía a lo largo de este proceso. A los profesores que me inculcaron con su ejemplo los deseos de ser un buen profesional, en especial a Tatiana, Yadilka, Irán, Isyed y Yasirys.

Por último deseo agradecer a la Universidad de las Ciencias Informáticas, a ella debo el haber conocido a todos esos amigos que llegarían a convertirse en parte de mi familia, y que me harían vivir los que han sido, los mejores años de mi vida.

Resumen

La correcta interacción de los usuarios con los sistemas informáticos está sustentada grandemente por la creación de interfaces gráficas que simplifiquen el trabajo con los mismos. En el caso del LMS Moodle, también se hace uso de estas interfaces gráficas, las cuales son conocidas como temas web. Para la creación de dichos temas web se utilizan algunas herramientas especializadas, las cuales agilizan el trabajo y suelen reducir el tiempo y esfuerzo que se emplea en el proceso. En el contexto actual de la universidad, se ve dificultada la utilización de estas aplicaciones, debido a que en su mayoría son privativas, incompatibles con los sistemas operativos empleados en la universidad, y no permiten exportar los temas creados a la versión 2.8 de Moodle empleada actualmente en la universidad.

En el presente trabajo se describe una herramienta informática destinada a facilitar el proceso de creación de temas web para el LMS Moodle en su versión 2.8. El resultado obtenido es una alternativa soberana y multiplataforma a las aplicaciones propietarias empleadas en el proceso de creación de temas web para Moodle; dicha herramienta está acorde a la migración llevada a cabo por la Universidad de las Ciencias Informáticas hacia sistemas operativos libres basados en GNU/Linux.

Palabras clave: tema web, Moodle, Linux

Índice

Resumen	6
Introducción	1
Capítulo 1: Fundamentación teórica	7
1.1 Introducción	¡Error! Marcador no definido.
1.2 Principales conceptos asociados a la investigación.....	7
1.2.1 Interfaz de usuario.....	7
1.2.2 Temas Web	9
1.2.3 Sistemas Gestores de Contenido (CMS)	9
1.2.4 Sistemas para la Gestión del Aprendizaje (LMS)	9
1.2.5 Moodle	10
1.3 Soluciones similares.....	10
1.4 Metodología de desarrollo	15
1.5 Tecnologías y herramientas.....	20
1.5.1 Herramientas CASE.....	20
1.5.2 Unified Modeling Language (UML).....	21
1.5.3 Lenguajes.....	21
1.5.4 Framework de desarrollo	24
1.5.5 Entornos de Desarrollo Integrado	24
1.5.6 Otras herramientas utilizadas	26
1.6 Conclusiones del capítulo	26
Capítulo 2: Solución propuesta	27
2.1 Introducción	¡Error! Marcador no definido.
2.2 Descripción de la solución propuesta.....	27
2.3 Modelo conceptual.....	27
2.3.1 Descripción de los conceptos	28
2.5 Requisitos del sistema	28
2.5.1 Requisitos funcionales del sistema.....	29
2.5.2 Requisitos no funcionales del sistema.....	30
2.6 Historias de usuario	30
2.7 Análisis y diseño de la solución propuesta	33

2.7.1 Diagrama de clases del análisis	33
2.7.2 Diagramas de colaboración del análisis.....	34
2.7.3 Arquitectura.....	35
2.7.4 Diagrama de clases del diseño	38
2.7.5 Diagramas de secuencia del diseño.....	40
2.7.6 Diagrama de despliegue	42
2.8 Conclusiones del capítulo	42
Capítulo 3: Implementación y pruebas	43
3.1 Introducción	¡Error! Marcador no definido.
3.2 Diagrama de componentes	43
3.3 Estándares de codificación	44
3.4 Pruebas de software.....	44
3.4.1 Niveles de prueba.....	44
3.4.2 Técnicas de prueba.....	45
3.4.4 Resultados de las pruebas de software.....	48
3.6 Conclusiones del capítulo	51
Conclusiones generales	52
Recomendaciones	53
Bibliografía.....	54
Anexo 1: Historias de usuario	¡Error! Marcador no definido.
Anexo 2: Diagramas de colaboración del análisis	¡Error! Marcador no definido.
Anexo 3: Diagramas de secuencia del diseño	¡Error! Marcador no definido.
Anexo 4: Casos de prueba	¡Error! Marcador no definido.

Índice de tablas

Tabla 1 Comparativa de metodologías ágiles y tradicionales(26)	15
Tabla 2 Requisitos funcionales del sistema	29
Tabla 3 Requisitos no funcionales del sistema	30
Tabla 4 HU4: Editar fondo general	31
Tabla 5 CP Editar fondo general	46
Tabla 6: No conformidades identificadas	48
Tabla 7: HU1 Exportar tema web	¡Error! Marcador no definido.
Tabla 8: HU2 Editar sombra general	¡Error! Marcador no definido.
Tabla 9: HU3 Editar tipografía general	¡Error! Marcador no definido.
Tabla 10: HU4 Editar fondo general	¡Error! Marcador no definido.
Tabla 11: HU5 Editar botones	¡Error! Marcador no definido.
Tabla 37: CP Exportar tema web	¡Error! Marcador no definido.
Tabla 38 CP Editar sombra general	¡Error! Marcador no definido.
Tabla 39: CP Editar estilo de los submenús	¡Error! Marcador no definido.
Tabla 40: CP Editar íconos	¡Error! Marcador no definido.

Índice de ilustraciones

Ilustración 1: Modelo conceptual del dominio.....	28
Ilustración 2: Diagrama de clases del análisis.....	34
Ilustración 3: Diagrama de clases del análisis (RF1 Exportar tema web)	34
Ilustración 4 DCA_HU4: Editar fondo general - escenario 1.....	34
Ilustración 5 DCA_HU4: Editar fondo general - escenario 2.....	35
Ilustración 6: Patrón Modelo-Vista-Modelo de vista	36
Ilustración 7: Ejemplo de uso del patrón creador	37
Ilustración 8: Ejemplo de uso del patrón controlador.....	38
Ilustración 9 Diagrama de clases del diseño.....	39
Ilustración 10 Diagrama de clases del diseño (RF1 Exportar tema web)	40
Ilustración 11 DSD_HU4: Editar fondo general – escenario 1	41
Ilustración 12 DSD_HU4: Editar fondo general – escenario 2	41
Ilustración 13: Diagrama de despliegue	42
Ilustración 14 Diagrama de componentes del sistema	43
Ilustración 15: DCA_HU1 Exportar tema web	¡Error! Marcador no definido.
Ilustración 16: DCA_HU2 Editar sombra general	¡Error! Marcador no definido.
Ilustración 17: DCA_HU3 Editar tipografía general	¡Error! Marcador no definido.
Ilustración 18: DCA_HU4 Editar fondo general - escenario 1.....	¡Error! Marcador no definido.
Ilustración 19: DCA_HU4 Editar fondo general - escenario 2.....	¡Error! Marcador no definido.
Ilustración 20: DCA_HU5 Editar botones - escenario 1.....	¡Error! Marcador no definido.
Ilustración 21: DCA_HU5 Editar botones - escenario 2.....	¡Error! Marcador no definido.
Ilustración 22: DCA_HU5 Editar botones - escenario 3.....	¡Error! Marcador no definido.
Ilustración 23: DCA_HU6 Editar íconos	¡Error! Marcador no definido.
Ilustración 24: DCA_HU7 Editar fondo de barra de navegación - escenario 1.....	¡Error! Marcador no definido.
Ilustración 25: DCA_HU7 Editar fondo de barra de navegación - escenario 2.....	¡Error! Marcador no definido.
Ilustración 26: DCA_HU8 Editar tipografía de barra de navegación	¡Error! Marcador no definido.
Ilustración 27: DCA_HU9 Editar sombra de barra de navegación.....	¡Error! Marcador no definido.
Ilustración 28: DCA_HU10 Editar bordes de barra de navegación.....	¡Error! Marcador no definido.
Ilustración 29: DCA_HU11 Editar estilo de los submenús	¡Error! Marcador no definido.
Ilustración 30: DCA_HU12 Editar sombra de los submenús	¡Error! Marcador no definido.
Ilustración 31: DCA_HU13 Editar tipografía de los submenús	¡Error! Marcador no definido.
Ilustración 32: DCA_HU14 Editar bordes de los submenús	¡Error! Marcador no definido.
Ilustración 33: DCA_HU15 Editar fondo de banner - escenario 1	¡Error! Marcador no definido.
Ilustración 34: DCA_HU15 Editar fondo de banner - escenario 2	¡Error! Marcador no definido.
Ilustración 35: DCA_HU17 Editar tipografía de banner.....	¡Error! Marcador no definido.
Ilustración 36: DCA_HU18 Editar bordes de banner	¡Error! Marcador no definido.
Ilustración 37: DCA_HU19 Editar fondo del deslizador - escenario 1	¡Error! Marcador no definido.
Ilustración 38: DCA_HU19 Editar fondo del deslizador - escenario 2	¡Error! Marcador no definido.
Ilustración 39: DCA_HU20 Editar tipografía del deslizador.....	¡Error! Marcador no definido.
Ilustración 40: DCA_HU21 Editar sombra del deslizador	¡Error! Marcador no definido.
Ilustración 41: DCA_HU22 Editar enlaces de bloques - escenario 1	¡Error! Marcador no definido.

Ilustración 42: DCA_HU22 Editar enlaces de bloques - escenario 2	¡Error! Marcador no definido.
Ilustración 43: DCA_HU22 Editar enlaces de bloques - escenario 3	¡Error! Marcador no definido.
Ilustración 44: DCA_HU23 Editar fondo de bloques - escenario 1	¡Error! Marcador no definido.
Ilustración 45: DCA_HU23 Editar fondo de bloques - escenario 2	¡Error! Marcador no definido.
Ilustración 46: DCA_HU24 Editar tipografía de bloques	¡Error! Marcador no definido.
Ilustración 47: DCA_HU25 Editar sombra de bloques	¡Error! Marcador no definido.
Ilustración 48: DCA_HU26 Editar bordes de bloques	¡Error! Marcador no definido.
Ilustración 49: DCA_HU27 Editar fondo de pie de página - escenario 1	¡Error! Marcador no definido.
Ilustración 50: DCA_HU27 Editar fondo de pie de página - escenario 2	¡Error! Marcador no definido.
Ilustración 51: DCA_HU28 Editar tipografía de pie de página	¡Error! Marcador no definido.
Ilustración 52: DCA_HU29 Editar enlaces de pie de página - escenario 1	¡Error! Marcador no definido.
Ilustración 53: DCA_HU29 Editar enlaces de pie de página - escenario 2	¡Error! Marcador no definido.
Ilustración 54: DCA_HU29 Editar enlaces de pie de página - escenario 3	¡Error! Marcador no definido.
Ilustración 55: DCA_HU30 Editar bordes de pie de página	¡Error! Marcador no definido.
Ilustración 56: DSD_HU1 Exportar tema web	¡Error! Marcador no definido.
Ilustración 57: DSD_HU2 Editar sombra general	¡Error! Marcador no definido.
Ilustración 58: DSD_HU3 Editar tipografía general	¡Error! Marcador no definido.
Ilustración 59: DSD_HU4 Editar fondo general - escenario 1	¡Error! Marcador no definido.
Ilustración 60: DSD_HU4 Editar fondo general - escenario 2	¡Error! Marcador no definido.
Ilustración 61: DSD_HU5 Editar botones - escenario 1	¡Error! Marcador no definido.
Ilustración 62: DSD_HU5 Editar botones - escenario 2	¡Error! Marcador no definido.
Ilustración 63: DSD_HU5 Editar botones - escenario 3	¡Error! Marcador no definido.
Ilustración 64: DSD_HU6 Editar íconos	¡Error! Marcador no definido.
Ilustración 65: DSD_HU7 Editar fondo de barra de navegación - escenario 2	¡Error! Marcador no definido.
Ilustración 66: DSD_HU7 Editar fondo de barra de navegación - escenario 2	¡Error! Marcador no definido.
Ilustración 67: DSD_HU8 Editar tipografía de barra de navegación	¡Error! Marcador no definido.
Ilustración 68: DSD_HU9 Editar sombra de barra de navegación	¡Error! Marcador no definido.
Ilustración 69: DSD_HU10 Editar bordes de barra de navegación	¡Error! Marcador no definido.
Ilustración 70: DSD_HU11 Editar estilo de los submenús	¡Error! Marcador no definido.
Ilustración 71: DSD_HU12 Editar sombra de los submenús	¡Error! Marcador no definido.
Ilustración 72: DSD_HU13 Editar tipografía de los submenús	¡Error! Marcador no definido.
Ilustración 73: DSD_HU14 Editar bordes de los submenús	¡Error! Marcador no definido.
Ilustración 74: DSD_HU15 Editar fondo de banner - escenario 1	¡Error! Marcador no definido.
Ilustración 75: DSD_HU15 Editar fondo de banner - escenario 2	¡Error! Marcador no definido.
Ilustración 76: DSD_HU17 Editar tipografía de banner	¡Error! Marcador no definido.
Ilustración 77: DSD_HU18 Editar bordes de banner	¡Error! Marcador no definido.
Ilustración 78: DSD_HU19 Editar fondo del deslizador - escenario 1	¡Error! Marcador no definido.
Ilustración 79: DSD_HU19 Editar fondo del deslizador - escenario 2	¡Error! Marcador no definido.
Ilustración 80: DSD_HU20 Editar tipografía del deslizador	¡Error! Marcador no definido.
Ilustración 81: DSD_HU21 Editar sombra del deslizador	¡Error! Marcador no definido.
Ilustración 82: DSD_HU22 Editar enlaces de bloques - escenario 1	¡Error! Marcador no definido.
Ilustración 83: DSD_HU22 Editar enlaces de bloques - escenario 2	¡Error! Marcador no definido.
Ilustración 84: DSD_HU22 Editar enlaces de bloques - escenario 3	¡Error! Marcador no definido.

Ilustración 85: DSD_HU23 Editar fondo de bloques - escenario 1 ¡Error! Marcador no definido.

Ilustración 86: DSD_HU23 Editar fondo de bloques - escenario 2 ¡Error! Marcador no definido.

Ilustración 87: DSD_HU24 Editar tipografía de bloques..... ¡Error! Marcador no definido.

Ilustración 88: DSD_HU25 Editar sombra de bloques ¡Error! Marcador no definido.

Ilustración 89: DSD_HU26 Editar bordes de bloques ¡Error! Marcador no definido.

Ilustración 90: DSD_HU27 Editar fondo de pie de página - escenario 1 ¡Error! Marcador no definido.

Ilustración 91: DSD_HU27 Editar fondo de pie de página - escenario 2 ¡Error! Marcador no definido.

Ilustración 92: DSD_HU28 Editar tipografía de pie de página..... ¡Error! Marcador no definido.

Ilustración 93: DSD_HU29 Editar enlaces de pie de página - escenario 1..... ¡Error! Marcador no definido.

Ilustración 94: DSD_HU29 Editar enlaces de pie de página - escenario 2..... ¡Error! Marcador no definido.

Ilustración 95: DSD_HU29 Editar enlaces de pie de página - escenario 3..... ¡Error! Marcador no definido.

Ilustración 96: DSD_HU30 Editar bordes de pie de página ¡Error! Marcador no definido.

Introducción

La evolución que ha experimentado internet desde sus orígenes es algo evidente, es difícil entonces no notar el surgimiento de conceptos como el de Web 2.0. Este término no hace alusión a una nueva tecnología como se pudiera pensar, sino a una nueva manera de concebir la web(1). La Web 2.0 establece una distinción entre su predecesor Web 1.0, pues está pensada para permitir procesos de interactividad contributiva, es decir, que el usuario pueda añadir y compartir información con otros usuarios(2), enriqueciendo por tanto la experiencia de navegación.

Con esta nueva tendencia de participación de los usuarios, lograr que la información se muestre de forma comprensible y amena se vuelve una tarea cada vez más complicada, pues cada vez se incluye una mayor variedad y cantidad de contenido en la web. Las páginas web de hoy poco tienen que ver con las de antaño, el avance de las tecnologías web ha propiciado que el contenido de las mismas se diversifique. Un sitio web en la actualidad incluye entre sus páginas contenido multimedia variado como son: imágenes, videos, audio, animaciones, texto, entre otros. Debido a la gran cantidad de contenido que existe en la red, puede verse dificultado el proceso de búsqueda de información, por lo que se hace necesario estructurarlo de manera correcta, pues el usuario podría llegar a sentirse desorientado si el contenido del sitio web no le es presentado de una manera accesible. Es en este proceso de hacer llegar la información al usuario donde entran a jugar su papel las interfaces gráficas de usuario.

Una interfaz gráfica de usuario (GUI por sus siglas en inglés) es la encargada de realizar la función de mediadora en el proceso interactivo humano-computadora(3). Su labor es garantizar que la información que se quiere mostrar llegue al usuario con la mayor calidad posible. Además, define en gran medida el éxito o no de un sitio web pues es capaz de crear sentimientos de fidelidad por parte de los usuarios hacia el producto.

Para la creación de sitios web es común utilizar Sistemas Gestores de Contenido (CMS de aquí en adelante). Un CMS es un software que permite la creación y administración de un sitio web de manera automática(4). De manera similar, un Sistema para la Gestión del Aprendizaje (LMS de aquí en adelante) posibilita la distribución de cursos en línea de instituciones u organizaciones, permitiendo así el aprendizaje electrónico(5). Estas herramientas brindan a los desarrolladores y usuarios grandes posibilidades de configuración de la apariencia, pues esta puede ser modificada casi instantáneamente mediante interfaces gráficas prediseñadas denominadas temas web.

El desarrollo del internet y la utilización de los LMS, han abierto un sin número de posibilidades para realizar proyectos educativos en el que todas las personas tengan la oportunidad de acceder a educación de calidad sin importar el momento o el lugar en el que se encuentren. Las alternativas de acceso que se han puesto en manos de las personas han eliminado el tiempo y la distancia como un obstáculo para enseñar y aprender. La educación en línea, es una modalidad de la educación a distancia, que utiliza internet junto con las tecnologías de la información y las comunicaciones para realizar el proceso de enseñanza-aprendizaje(6).

La Universidad de las Ciencias Informáticas (UCI de aquí en adelante), es una institución cubana creada con la misión de "(...) formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática. Producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación. Servir de soporte a la industria cubana de la informática"¹.

Entre los centros que conforman la estructura productiva de la UCI se encuentra el Centro de Tecnologías para la Formación (FORTES de aquí en adelante), cuya misión es "(...) desarrollar tecnologías que permitan ofrecer servicios y productos para la implementación de soluciones de formación aplicando las Tecnologías de la Información y las Comunicaciones, a todo tipo de instituciones con diferentes modelos de formación y condiciones tecnológicas, garantizando la calidad de las soluciones y la formación de los recursos humanos a partir de investigaciones que combinen los elementos pedagógicos y tecnológicos más avanzados, integrando así los procesos de formación, producción e investigación"².

La UCI, como institución docente-productiva, hace uso de plataformas educativas para la creación de cursos en línea. El uso de estas nuevas tendencias permite a alumnos y docentes llevar la educación a un nivel más alto, los encuentros presenciales ahora son respaldados por las aulas virtuales. Los alumnos podrán tener acceso a todo un cúmulo de información y materiales didácticos puestos a su disposición con el objetivo de facilitar el aprendizaje. Los profesores ya no verán al aula como único modo para impartir los conocimientos, ahora podrán hacer llegar a sus estudiantes todo el conocimiento haciendo uso de una vía que resulta interesante y atractiva, Internet(9).

¹ Tomado del sitio oficial de la Universidad de las Ciencias Informáticas(7).

² Tomado del sitio oficial de la Facultad 4 de la Universidad de las Ciencias Informáticas(8).

Para la creación de los cursos en línea que se utilizan en la universidad, es ampliamente empleado el LMS Moodle. La personalización de esta plataforma es una labor de vital importancia, pues con esto, se logra un mayor acercamiento entre los usuarios y estos sitios creados con propósitos educativos, es necesario confeccionar interfaces que capturen la atención del estudiante haciéndole que se interese cada vez más por el contenido presentado y de esta manera lograr que se cumpla el objetivo que se persigue con estos entornos educativos, el cual es lograr hacer llegar el conocimiento al usuario. Esta personalización es llevada a cabo por el centro FORTES mediante la creación de temas web personalizados.

Actualmente esta labor se realiza de manera manual, es necesario crear un directorio que contendrá la estructura de carpetas requeridas para el correcto funcionamiento del tema que se va a crear. Se deberán crear y configurar todos los ficheros que lo compondrán y establecer todas las dependencias hacia los temas base de los cuales extiende el tema que se va a crear. Además, es necesario contar con conocimientos medios o avanzados de programación y de diseño web. Los desarrolladores han planteado su inconformidad con este proceso, pues realizar esta tarea lleva implícito un gran empleo de tiempo y esfuerzo.

De todo lo previamente planteado se deriva como **problema a resolver**: ¿Cómo facilitar el proceso de creación de temas web para la versión 2.8 de la plataforma Moodle?

Para dar solución al problema previamente planteado se define como **objetivo general** de la investigación: desarrollar una herramienta multiplataforma que facilite la creación de temas web para la versión 2.8 de la plataforma Moodle.

Se define además como **objeto de estudio** de la investigación el proceso de desarrollo de temas web. Se centra el **campo de acción** de la misma en el proceso de desarrollo de temas web para el LMS Moodle.

Para dar cumplimiento al objetivo general se definieron los siguientes **objetivos específicos**:

- Establecer las bases teóricas que sustentan la creación de temas web para la plataforma Moodle.
- Realizar el análisis y diseño de la herramienta.
- Implementar la herramienta, de forma que cumpla con los requisitos definidos.
- Validar la herramienta desarrollada mediante pruebas de software.

Para alcanzar los objetivos propuestos se concibieron las siguientes **preguntas científicas**:

- ¿Qué plantea la bibliografía especializada sobre el objeto y el campo de acción de la investigación?
- ¿Cuál es el estado actual del proceso de desarrollo de temas web para la plataforma Moodle?
- ¿Cómo desarrollar una herramienta que facilite la creación de temas web compatibles con las versiones de la plataforma Moodle empleadas en el centro FORTES?
- ¿Qué resultados se obtienen con la utilización de esta herramienta?

Para elaborar el trabajo de investigación se proponen las siguientes **tareas de investigación**:

- Revisión de diferentes fuentes bibliográficas para conocer qué se ha investigado sobre el tema.
- Estudio del estado actual del proceso de desarrollo de temas web para el LMS Moodle y las herramientas existentes utilizadas para ese fin.
- Análisis del funcionamiento y estructura de los temas web y su integración con Moodle.
- Análisis comparativo de las herramientas similares existentes.
- Selección de la metodología, tecnologías y herramientas adecuadas para el desarrollo del producto.
- Identificación de los requisitos funcionales y no funcionales de la solución.
- Diseño de la solución en correspondencia con la metodología seleccionada.
- Implementación de las funcionalidades de la propuesta de solución.
- Validación de la herramienta mediante pruebas de software.

El **resultado esperado** de la presente investigación es el siguiente:

Una herramienta multiplataforma que permita la creación de temas web para la versión 2.8 de la plataforma Moodle.

En el transcurso de la investigación se emplean un conjunto de métodos científicos los cuales se definen a continuación:

Métodos Teóricos:

Analítico-Sintético: es empleado para analizar y sintetizar la información obtenida durante el estudio de los principales conceptos asociados al proceso de creación de temas web para la plataforma Moodle, las interfaces de usuarios, los temas para los sistemas gestores de contenido y las herramientas similares existentes.

Histórico-Lógico: se utiliza en el análisis de la evolución de las herramientas para la creación de temas web y además para poder entender con claridad el estado real de los conceptos asociados al proceso de generación de temas web para la plataforma Moodle en el momento histórico en que se desarrolla la investigación.

Métodos Empíricos:

Observación: se utiliza este método para obtener información relacionada con el estado actual del proceso de creación de temas web para la plataforma Moodle.

Estructura Capitular:

La presente investigación se estructura con 3 capítulos distribuidos de la manera siguiente:

Capítulo 1: Fundamentación teórica

Contiene los fundamentos teóricos que sostienen la investigación. Se definen además cuáles serán las herramientas y metodologías a utilizar para el desarrollo de la solución.

Capítulo 2: Propuesta de solución

En este capítulo se exponen las características generales de la propuesta de solución y se realiza la captura de los requisitos funcionales y no funcionales de la misma. Se elaboran las historias de usuarios necesarias para la descripción de dichos requisitos, y se realiza el análisis y diseño que contribuirá a simplificar la posterior implementación de la herramienta.

Capítulo 3: Implementación y prueba

Este capítulo aborda la fase de implementación de las funcionalidades de la solución y la conclusión del ciclo de desarrollo con la realización de pruebas de software para validar la solución.

Capítulo 1: Fundamentación teórica

Una herramienta para la generación de temas web consiste en un software que mediante interfaces intuitivas permite a los usuarios crear temas web mientras contribuye a la reducción del tiempo y el esfuerzo empleado en el proceso. Para una mejor comprensión de la presente investigación se realiza un estudio de los conceptos fundamentales relacionados con la misma. Se hace necesario la realización de un análisis de las soluciones similares existentes con el fin de identificar una herramienta que brinde solución al problema planteado en la investigación o determinar si es necesario la implementación de una herramienta propia; en cuyo caso es de importancia la selección de herramientas, tecnologías y metodología a emplear en el proceso de desarrollo de la investigación.

1.1 Principales conceptos asociados a la investigación

A continuación, se describen algunos de los conceptos asociados a la presente investigación, la comprensión y el conocimiento de los mismos es de gran importancia para el desarrollo de la propuesta de solución y para el entendimiento de la investigación.

1.1.1 Interfaz de usuario

Las interfaces básicas de usuario son aquellas que incluyen elementos como menús, ventanas, teclado, ratón, sonidos que la computadora hace, en general, todos aquellos canales por los cuales se permite la comunicación entre el hombre y la computadora(10). En computación, la interfaz de usuario es el espacio por medio del cual se pueden comunicar las personas con las máquinas para que así los usuarios puedan operar y controlar a la máquina, y que esta a su vez envíe retroalimentación para ayudar al operador a tomar decisiones y realizar tareas(11).

De manera más técnica se pudiera decir que una interfaz de usuario es el conjunto de componentes que intervienen en el proceso de interacción hombre-máquina y que posibilita que un usuario común, de manera intuitiva, pueda dirigir el funcionamiento de la máquina, indicarle las operaciones que debe realizar y una vez completadas estas peticiones apreciar los resultados obtenidos.

A modo de resumen, y vinculándose con el tema que se trata en esta investigación, se puede definir el término interfaz de usuario como la parte de una aplicación que el usuario ve y con la cual interactúa. La interfaz incluye ventanas, controles, menús, la ayuda en línea, la documentación y el entrenamiento. Cualquier cosa que el usuario vea y con lo cual interactúe es parte de la interfaz.

Con el avance del tiempo, las interfaces de usuario han evolucionado, modificando su aspecto visual y su forma de interacción. A continuación, se describen algunas clasificaciones de las interfaces de usuario teniendo en cuenta su modo de interacción con los usuarios.

Interfaz de menú

Una interfaz de menú permite que el usuario elija las posibles opciones de una lista en pantalla. Al responder el usuario se ve limitado por las opciones que se le presentan en el menú. El usuario no necesita conocer el sistema pero sí necesita saber qué tareas pueden realizarse. Sin embargo, para sacar el máximo provecho del menú, los usuarios deben estar al tanto de las tareas potenciales del sistema y de lo que realmente desean hacer(12).

Interfaz de preguntas y respuestas

En este tipo de interfaces la computadora plantea sobre la pantalla una pregunta al usuario. Con el fin de interactuar, el usuario por lo general proporciona una respuesta (a través de un teclado) y la computadora responderá con base en tal información de entrada de una manera pre-programada, después de ello, el cursor se desplazará a la siguiente pregunta, y así sucesivamente(12).

Interfaz Gráfica de Usuario (GUI)

Las interfaces gráficas de usuario son la cara visible de los ordenadores y también de otros tipos de dispositivos electrónicos. Ayudan al usuario a interactuar con la máquina, expanden el rango de aplicaciones de una computadora de forma considerable y representan una importante ayuda para el aprendizaje del trabajo con ordenadores. Además, no hay que menospreciar la importancia que han tenido en la popularización de la informática fuera de ámbitos corporativos y científicos, al reducir la cantidad de conocimiento acerca de las máquinas necesario para un uso eficaz, práctico y útil de las mismas(13).

Tomando en consideración las definiciones abordadas previamente y las características propias de cada tipo de interfaz, se decide que en la presente investigación se hará uso específicamente del concepto de Interfaz Gráfica de Usuario (GUI por sus siglas en inglés), debido a la relación que guarda con el tópico tratado en el trabajo.

1.1.2 Temas Web

Un tema web es una colección de archivos relacionados entre sí que interactúan con el núcleo de un CMS y son responsables de la apariencia final del sitio web generado. Definen algunos aspectos visuales globales tales como la combinación de colores, los tipos de letra y la composición de las páginas. Además determina la distribución de los elementos como controles para la navegación, información fija, información variable así como los elementos gráficos comunes a todas las páginas como íconos, botones, barras de navegación o imágenes de fondo(14).

1.1.3 Sistemas Gestores de Contenido (CMS)

Un Sistema de Gestión de Contenido-CMS (Content Management System, en inglés) es un software que permite la creación y administración de los contenidos de una página web, principalmente, de forma automática. Así, con él es posible publicar, editar, borrar, otorgar permisos de acceso o establecer los módulos visibles para el visitante final de la página.

En definitiva, cuando se habla de un CMS se hace referencia a una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. Las características de un CMS pueden variar, pero la mayoría incluye publicación basada en Web, indexación, revisión, búsqueda y recuperación de la información(4).

El sistema permite manejar de manera independiente el contenido por una parte y el diseño por otra. De esta manera, es posible manejar el contenido y variar en cualquier momento el diseño del sitio sin tener que darle formato a todo el contenido de nuevo.

Además, permite de manera fácil y controlada la publicación en el sitio a varios editores para que éstos puedan escribir artículos, proponer votaciones, definir la apariencia y el funcionamiento de las interfaces gráficas del sitio, entre otros aspectos(4).

1.1.4 Sistemas para la Gestión del Aprendizaje (LMS)

Un sistema de gestión de aprendizaje-LMS (Learning Management System, en inglés) es un software que automatiza la administración de acciones de formación(4). Entre las funcionalidades con las que cuentan los LMS se encuentran las siguientes:

- Registra a todos los actores que intervienen en el acto de aprendizaje (alumnos, profesores, administradores).

- Organiza los diferentes cursos en un catálogo, almacena datos sobre los usuarios, realiza un seguimiento del aprendizaje y la temporización de los trámites y genera informes automáticamente para tareas de gestión específicas.
- La mayoría de los LMS están basados en la Web para facilitar en cualquier momento, en cualquier lugar y a cualquier ritmo el acceso a los contenidos de aprendizaje y administración.

En definitiva, se puede decir que un LMS puede considerarse como un CMS de propósito específico, concretamente educativo, que potencia las posibilidades de colaboración e interactividad que puede ofrecer un espacio virtual(4).

1.1.5 Moodle

Moodle es una plataforma de aprendizaje diseñada para proporcionarles a educadores, administradores y estudiantes un sistema integrado único, robusto y seguro para crear ambientes de aprendizaje personalizados. El número de usuarios de Moodle a nivel mundial, de más de 79 millones de usuarios, entre usuarios académicos y empresariales, lo convierten en la plataforma de aprendizaje más ampliamente utilizada del mundo. Una interfaz simple, funcionalidad de arrastrar y soltar, y recursos bien documentados, junto con mejoras continuas en usabilidad, hacen a Moodle fácil de aprender y usar.

Moodle es proporcionado gratuitamente como programa de código abierto, bajo la Licencia Pública General GNU (GNU General Public License). Por lo tanto, cualquier persona puede adaptar, extender o modificar Moodle(15).

1.2 Soluciones similares

El auge que toma en la actualidad el uso de Sistemas Gestores de Contenido para la fácil y rápida generación de sitios web y la importancia que se le asigna al diseño web para obtener productos que capten la atención de los usuarios, ha traído consigo la aparición de herramientas que contribuyen a hacer del proceso de creación de plantillas o temas web una tarea simple y amena. Estas herramientas abren las puertas a usuarios con poca experiencia en el diseño web pues la mayoría de las mismas cuentan con interfaces intuitivas y sencillas de utilizar.

A continuación se procede a analizar algunas de estas herramientas para la generación de temas web personalizados.

TemplateToaster

TemplateToaster es un software de diseño web para la creación de plantillas y temas web. Brinda soporte para exportar las plantillas creadas a los CMS WordPress, Drupal, Magento y Joomla. Los temas web creados con TemplateToaster se mostrarán correctamente en cualquier dispositivo sin importar el tamaño de la pantalla, pues hace uso de Diseño Web Sensible (RWD por sus siglas en inglés).

El trabajo con esta herramienta resulta bien sencillo gracias a su funcionalidad Arrastrar y Soltar, es posible crear un sitio web desde cero solamente arrastrando, posicionando y redimensionando los elementos al gusto, por lo cual puede ser utilizada tanto por usuarios sin grandes conocimientos del desarrollo web como por usuarios experimentados. A las características previamente mencionadas se suman otras que hacen muy interesante el trabajo con TemplateToaster(16):

- Posee un editor WYSIWYG (What You See Is What You Get)³ avanzado para la edición visual.
- Posee una biblioteca de imágenes y esquemas de colores incluidos.
- Permite realizar operaciones de escalado y recortado de imágenes.
- Cuenta con un generador para la creación de cualquier tipo de Menú.
- Cientos de fuentes disponibles, incluido soporte para Google Fonts.
- Exporta archivos HTML⁴ y CSS⁵ acordes con los estándares de W3C⁶.
- Soporte para las últimas liberaciones de HTML5 y CSS3.
- Integración con redes sociales.
- Incorpora el Protocolo de Transferencia de Archivos (FTP por sus siglas en inglés).

Sin lugar a dudas TemplateToaster es una excelente herramienta para la creación de temas web pero su utilización se ve limitada por algunos inconvenientes. Es un software propietario por lo que se hace necesario efectuar un pago por su licencia de uso y además, no se puede utilizar en sistemas basados en GNU/Linux pues solo se encuentra disponible en su versión para Microsoft Windows.

³ En español (Lo que ves es lo que obtienes), se aplica a los editores de texto con formato, permitiendo observar el resultado final según se va modificando el documento.

⁴ HyperText Markup Language (Lenguaje de Marcas de Hipertexto). Lenguaje de marcado usado para la elaboración de páginas web.

⁵ Cascading Style Sheets (Hojas de Estilo en Cascada). Lenguaje empleado para definir el diseño o estilo de una página web.

⁶ World Wide Web Consortium (Consortio Mundial de la Red). Es un consorcio internacional que produce estándares que aseguran el crecimiento de la red a largo plazo.

Artisteer

Artisteer es una aplicación de escritorio para la personalización de sitios web mediante la creación de plantillas. Permite la creación de fantásticos temas web en cuestión de minutos sin necesidad de tener conocimiento de CSS, HTML o Photoshop. Las plantillas generadas con este software son soportadas por algunos de los CMS más utilizados actualmente a nivel mundial. Esta aplicación cuenta además con otras características entre las que cabe mencionar(17):

- Facilidad de pre-visualizar, descargar y modificar plantillas previamente diseñadas.
- Creación de plantillas para Blogger, Moodle, WordPress, Joomla y Drupal.
- Incluye elementos de diseño que van desde fondos hasta fotos y botones.
- Código HTML y CSS en conformidad con estándares web.
- Interfaz simple e intuitiva.
- Aplicación fácil de utilizar.

Aunque Artisteer no incluye de forma nativa el plugin que permite exportar plantillas para Moodle, sí es posible instalarlo sin costo alguno desde el Artisteer Marketplace. A pesar de esto, es una herramienta propietaria disponible solo para Microsoft Windows.

WordPress Theme Generator

WP Theme Generator facilita la creación de ilimitados temas web cambiando, combinando o cargando cualquier elemento de diseño en el tema, y una vez se obtenga el resultado deseado, guardarlo para una posterior edición o descargarlo para ser usado en WordPress como plantilla. Se encuentra en constante actualización y nuevas características son añadidas diariamente por el equipo de desarrollo incrementando los beneficios de usar esta herramienta. Entre sus características más notables se encuentran(18):

- Más de 300 temas prediseñados listos para descargar y usar.
- Usa Diseño Web Sensible, los temas generados lucirán bien en cualquier resolución de pantalla.
- Más de 1000 objetos de diseño puestos a la disposición del usuario.
- Más de 50 fuentes para elegir.
- Permite cargar diseños propios y usarlos en la creación de los temas.
- Disponible en varios idiomas.

No cabe duda de que WP Theme Generator es una potente herramienta a la hora de crear temas web, no obstante, los temas exportados por esta aplicación solo son compatibles con WordPress.

Gantry

Gantry es un framework usado para el desarrollo de temas para CMS rápido y eficiente, con muchas facilidades de uso para el usuario final. La creación de temas web con Gantry es una tarea sencilla e intuitiva, pues cuenta con un gestor de diseño que garantiza que este proceso de creación sea una experiencia visual. Por lo tanto, no es necesario que el usuario tenga experiencia en las tecnologías web para trabajar con esta herramienta. Además de lo previamente mencionado, Gantry cuenta con otras funcionalidades como son(19):

- Licencia libre de costo.
- Herramienta multiplataforma rápida y ligera.
- Posee un editor de menús para la configuración y estructuración de los mismos.
- Fácil creación de temas gracias a la funcionalidad arrastrar y soltar.
- Permite deshacer cualquier cantidad de cambios no deseados.
- Creación de temas usando el sistema de plantillas Twig⁷.

A pesar de las funcionalidades con las que cuenta Gantry, este presenta como deficiencia que solo exporta plantillas soportadas por Joomla y WordPress.

Soluciones nacionales

En la UCI, en aras de alcanzar un mayor desarrollo web, se ha incursionado en la creación de algunas aplicaciones para la creación de temas web.

Herramienta generadora de plantillas HTML

Liberada en el año 2011 por el autor Elvis Hernández Pérez. Es una herramienta multiplataforma y de código abierto, la cual permite la generación de plantillas HTML pero estas no son soportadas por ningún CMS(21).

TRAXOS

Herramienta multiplataforma liberada el 2015 por el autor Alberto Torres Junco. Distribuida bajo Licencia Publica General (GPL). Esta permite la creación de temas web para el LMS Moodle y algunos de los CMS más utilizados en la universidad. A pesar de esto, dicha herramienta no es compatible con las versiones de Moodle empleadas en el centro, no soporta la creación de temas web adaptativos acorde a las nuevas tecnologías web, y tampoco incluye tipografías propias que garanticen que los temas

⁷ Twig es un motor de plantillas para PHP, rápido, seguro y flexible(20).

generados mantengan su aspecto al cambiar de plataforma. Además, debido a que es una aplicación de escritorio, incurre en un gasto en memoria RAM y en espacio en disco duro, y no provee una completa integración con la plataforma Moodle(22).

Valoración de las herramientas existentes y aportes de la investigación

Luego de haber realizado una investigación sobre las herramientas existentes empleadas en el proceso de creación de temas web se pueden resaltar algunos rasgos de interés sobre las mismas:

- TemplateToaster: cuenta con una interfaz intuitiva, favorable para usuarios no familiarizados con el proceso de creación de temas web, pero es necesario efectuar un pago por su licencia de uso además de ser solo compatible con Microsoft Windows y no exportar sus plantillas para Moodle.
- Artisteer: esta herramienta permite instalar un plugin que posibilita exportar plantillas a Moodle, no obstante, es privativa y compatible solo con el sistema Windows.
- WordPress Theme Generator: posee un gran número de características que posibilitan la creación de ilimitados temas web, pero estos solo son compatibles con el CMS WordPress.
- Gantry: cuenta con una licencia gratuita además de ser multiplataforma, sin embargo, solo exporta temas soportados por Joomla y WordPress.
- La Herramienta Generadora de Plantillas HTML es de código abierto y multiplataforma, pero no posee la capacidad de exportar sus plantillas a ningún CMS.
- TRAXOS: aplicación multiplataforma y de código abierto, mas no es compatible con las últimas versiones de Moodle empleadas en el centro.

En consideración de los elementos analizados, se concluye que ninguna de las herramientas analizadas resuelve el problema planteado en la investigación. Se hace necesario entonces el desarrollo de una herramienta de código abierto que facilite la creación de temas web para la versión 2.8 de la plataforma Moodle y sea funcional tanto en Microsoft Windows como en sistemas basados en GNU/Linux. Se decide incorporar, además, como resultado del estudio realizado a las herramientas similares existentes un conjunto de características que resultarán ventajosas para el futuro desempeño de la propuesta de solución:

- Uso de diseño web sensible para la elaboración de los temas.
- Posibilidad de previsualizar y descargar plantillas diseñadas por la herramienta.
- Editor WYSIWYG incorporado para la edición visual.
- Interfaz sencilla y de fácil utilización.

1.3 Metodología de desarrollo

Una metodología de desarrollo de software es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información(23). Existen dos grandes enfoques en lo que respecta a las metodologías de desarrollo: el tradicional y el ágil.

Las metodologías tradicionales son aquellas que resultan convenientes de utilizar para el desarrollo de proyectos de gran envergadura. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Por lo general requieren de equipos de desarrollo grandes(24). Otra de las características importantes dentro de este enfoque son los altos costos al implementar un cambio y no ofrecer una buena solución para proyectos donde el entorno es volátil(25).

Por otra parte, las metodologías ágiles son apropiadas en pequeños proyectos y suelen requerir grupos de trabajo reducidos(24). Dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Estas metodologías muestran su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad(26).

La siguiente tabla muestra un resumen comparativo de ambos enfoques.

Tabla 1 Comparativa de metodologías ágiles y tradicionales(26)

Metodologías Ágiles	Metodologías Tradicionales
Pocos artefactos. El modelado es prescindible, modelos desechables.	Mayor número de artefactos. El modelado es esencial, mantenimiento de modelos.
Pocos roles, más genéricos y flexibles.	Mayor número de roles, más específicos.
No existe un contrato tradicional, debe ser bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Orientada a proyectos pequeños. Corta duración (o entregas frecuentes), equipos pequeños (menos de 10 integrantes) y trabajando en el mismo sitio.	Aplicable a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos.
La arquitectura se va definiendo y mejorando a lo	Se promueve que la arquitectura se defina

largo del proyecto.	tempranamente en el proyecto.
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo.	Énfasis en la definición del proceso: roles, actividades y artefactos.
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Se esperan cambios durante el proyecto.	Se espera que no ocurran cambios de gran impacto durante el proyecto

Como resultado del análisis comparativo previamente realizado y tomando en consideración que el proceso de desarrollo de la herramienta Moux será un proyecto de pequeña envergadura y de poca duración, con un equipo de desarrollo de pocos miembros, en el cual se necesita generar solo los artefactos necesarios de modo que se pueda realizar la documentación necesaria en el tiempo disponible para el proyecto, se decide utilizar una metodología de enfoque ágil. A continuación se describen las características de algunas de las metodologías de enfoque ágil más ampliamente utilizadas.

eXtreme Programming (XP)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico(26).

Entre las características principales de XP se encuentran(25):

- Su capacidad de programar de manera flexible las funcionalidades, respondiendo a las cambiantes necesidades del negocio.
- Dependencia de las pruebas automatizadas escritas por programadores y clientes para monitorear el progreso del desarrollo, para permitir que el sistema evolucione y para detectar tempranamente los errores.
- El cliente se integra como un miembro más al equipo de desarrollo.
- Se promueve una estrecha colaboración entre programadores con habilidades ordinarias(27).
- Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.

- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.

Scrum

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos(28). En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales(28).

Con la metodología Scrum un proyecto se ejecuta en bloques temporales (iteraciones-sprints) de un mes natural. Cada iteración tiene que proporcionar un resultado completo, un incremento de producto que sea susceptible de ser entregado con el mínimo esfuerzo cuando el cliente lo solicite(29).

El sprint es el ritmo de los ciclos de Scrum. Está delimitado por la reunión de planificación del sprint y la reunión retrospectiva. Una vez que se fija la duración del sprint es inamovible. Diariamente durante el sprint, el equipo realiza una reunión de seguimiento muy breve. Al final del sprint se entrega el producto al cliente en el que se incluye un incremento de la funcionalidad que tenía al inicio del sprint(29).

Kanban

Kanban es una metodología de enfoque ágil para el desarrollo de software creada por David Anderson. Su objetivo es gestionar de manera general como se van completando tareas, pero en los últimos años se ha utilizado en la gestión de proyectos de desarrollo software. Las principales reglas de Kanban son las siguientes(29, 30):

1. Visualizar el trabajo y las fases del ciclo de producción o flujo de trabajo.

- Dividir el trabajo en partes, escribir cada elemento en una tarjeta y pegarla en la pizarra.

Utilizar columnas nombradas para ilustrar en qué parte del flujo de trabajo se encuentra cada elemento.

2. Determinar el límite del trabajo en curso: asignar límites explícitos a cuantos elementos puedan estar en curso en cada estado del flujo de trabajo.

3. Medir el tiempo en completar una tarea: optimizar el proceso para hacer tiempo de espera tan pequeño y predecible como sea posible.

Algunos de los beneficios comúnmente observados al utilizar esta metodología son(30):

- Los cuellos de botella se hacen claramente visibles en tiempo real. Esto lleva a la gente a colaborar para optimizar toda la cadena de valor en lugar de solo su parte.

- Tiende a extenderse de forma natural en toda la organización a otros departamentos, como recursos humanos y ventas, lo que aumenta la visibilidad de todo lo que está pasando en la empresa.
- Reducción del tiempo perdido (todo el desarrollo se hace bajo demanda, por lo que no hay partes que sobren, además si alguien acaba su trabajo ayuda al resto del equipo).

Agile Unified Process (AUP)

El Proceso Unificado Ágil (AUP por sus siglas en inglés) es un híbrido creado por Scott Ambler, que combina el Proceso Unificado Racional (RUP por sus siglas en inglés) con métodos ágiles. AUP es un proceso iterativo incremental que se basa en flujos de trabajos y fases(31).

La Universidad de Ciencias Informáticas cuenta con numerosos centros productivos, y actualmente su actividad productiva se caracteriza por la utilización de diferentes metodologías, tanto de enfoque ágil como tradicional. No obstante se pudo comprobar que a pesar de utilizar una gran variedad de metodologías, estas no son aplicadas en su totalidad(32).

Se realiza entonces una variación a la metodología AUP de modo que se adapte al ciclo de vida definido para la actividad productiva UCI.

De las 4 fases propuestas por AUP (Inicio, Elaboración, Construcción, Transición), se decide para la variante UCI mantener la fase de Inicio modificando el objetivo de la misma, unificar las fases Elaboración, Construcción y Transición en una llamada Ejecución, y se agrega la fase Cierre(32).

Por otra parte, de las 7 disciplinas propuestas por AUP (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno) se plantea para el ciclo de vida de los proyectos de la UCI la utilización de 7 disciplinas también, pero subdivididas a un nivel más específico. Estas disciplinas son Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas internas, de liberación y de aceptación(32).

Fundamentación de la metodología seleccionada

Luego del estudio efectuado a algunas de las metodologías de enfoque ágil existentes y una vez analizadas sus principales características se decide utilizar AUP en su variante UCI para el desarrollo de la solución. Se selecciona dicha metodología porque está enfocada específicamente al proceso de desarrollo de software en la UCI. Debido a que es una metodología de enfoque ágil es favorable para proyectos de pequeña envergadura. Se centra en las actividades que son realmente importantes para el proceso de desarrollo. Toda la documentación es generada de manera simple y concisa evitando documentos innecesariamente largos. Sus roles y fases están bien definidas.

1.4 Tecnologías y herramientas

1.4.1 Herramientas CASE

Las herramientas CASE (Computer-aided software engineering) ayudan a los gerentes y profesionales de ingeniería de software en cada actividad asociada con el proceso de software. Automatizan las actividades de gestión de proyectos, gestionan todos los productos de trabajo generados en todo el proceso, y ayudan a los ingenieros en su trabajo de análisis, diseño, codificación y pruebas(33). Entre las herramientas CASE que se utilizan actualmente en el proceso de desarrollo de software se encuentran Visual Paradigm, Rational Rose Enterprise, EasyCASE, ArgoUML y otras muchas.

Visual Paradigm

Visual Paradigm para UML está diseñada para ayudar el desarrollo de software. Soporta lenguajes de modelado y estándares de la industria del software tales como Unified Modeling Language (UML), SysML, BPMN, XMI, entre otros. Ofrece un conjunto de funcionalidades que dan respuesta a todo un conjunto de necesidades como son realizar la captura de requisitos, planificación de software, planificación de pruebas, modelado de clases, modelado de datos, y otras tareas(34).

Entre las ventajas que trae consigo su utilización se encuentran(35):

- Perfecta integración con varios Entornos de Desarrollo Integrado (IDEs).
- Soporte Multiplataforma.
- Conversión instantánea de código fuente y archivos ejecutables en modelos UML y viceversa.
- Generación de documentación en formatos PDF, DOC, entre otros.

Rational Rose Enterprise

Rational Rose Enterprise proporciona un conjunto de prestaciones controladas por modelo para desarrollar aplicaciones de software, incluidas aplicaciones Ada, ANSI C++, CORBA, Java, Java EE, Visual C++ y Visual Basic. El software permite acelerar el desarrollo de estas aplicaciones con código generado a partir de modelos visuales mediante el lenguaje UML. Ofrece una herramienta y un lenguaje de modelado común para simplificar el entorno de trabajo y permitir una creación más rápida de software de calidad. Algunas de las características que presenta esta herramienta son(36):

- Proporciona prestaciones de modelado visual para desarrollar muchos tipos de aplicaciones de software.
- Contiene herramientas web y XML para el modelado de aplicaciones web.
- Integración del diseño de aplicaciones con el desarrollo: unifica el equipo del proyecto proporcionando una ejecución y una notación de modelos UML comunes.

ArgoUML

ArgoUML es un entorno gráfico interactivo, potente y fácil de usar, que soporta el diseño, desarrollo y documentación de aplicaciones de software orientadas a objeto. Entre sus principales características se encuentran(37):

- Estándares abiertos: XMI, SVG y PGML.
- Plataforma 100% independiente gracias al uso exclusivo de Java.
- Open Source, lo que posibilita su ampliación o personalización.
- Integrado con la web.
- Brinda una excelente ayuda.

Selección de la herramienta de modelado

De las herramientas para el modelado previamente analizadas se decide utilizar Visual Paradigm para UML 8.0 por las siguientes razones:

- Soporta todo el ciclo de vida del desarrollo de software.
- Es multiplataforma.
- Posee una versión gratuita, la Community Edition, la cual está libre de pagos de licencia.
- Es utilizada en los proyectos productivos de la Universidad de las Ciencias Informáticas.

1.4.2 Unified Modeling Language (UML)

UML es un lenguaje de modelado orientado a objetos que se ha convertido en un estándar para la representación y modelado de la información con la que se trabaja en las fases de análisis y, especialmente, de diseño (38). UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten. Otro objetivo de este modelado visual es que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos) (38).

1.4.3 Lenguajes

Para el desarrollo de la aplicación se hace necesario el empleo de varios lenguajes tanto del lado del servidor como del cliente, así como lenguajes de marcado⁸ y de diseño. Tomando en cuenta el ranking elaborado por TIOBE⁹, actualizado al mes de diciembre del 2015(21) se seleccionan y describen algunos

⁸ Un "Lenguaje de marcado" o "lenguaje de marcas" se puede definir como una forma de codificar un documento donde, junto con el texto, se incorporan etiquetas, marcas o anotaciones con información adicional relativa a la estructura del texto y su presentación(39).

⁹ TIOBE Software. Comunidad de Índice de Programación(40).

de los lenguajes para el desarrollo web más utilizados y de este modo elegir los lenguajes que se utilizarán para implementar la solución.

Lenguajes del lado del cliente

JavaScript

JavaScript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web o en programas mucho más grandes y complejos. Con JavaScript es posible crear diferentes efectos e interactuar con los usuarios. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos(41).

Este lenguaje posee varias características, entre ellas cabe mencionar que es un lenguaje basado en acciones. Gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, utilización de teclas, cargas de páginas, entre otros. JavaScript es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros(41).

Lenguajes del lado del servidor

PHP

PHP (acrónimo recursivo para PHP: Hypertext Preprocessor) es un lenguaje de script¹⁰ de código abierto, multipropósito, multiplataforma y ampliamente utilizado. Está especialmente equipado para el desarrollo web y puede ser embebido dentro del HTML de manera sencilla. PHP es un lenguaje que se ejecuta en el servidor, esto quiere decir que el cliente recibirá el HTML generado como resultado de la ejecución del código PHP procesado por el servidor sin necesidad de conocer el código subyacente. Aunque es extremadamente sencillo para principiantes brinda además estupendas posibilidades a programadores profesionales(43).

Python

Python es un lenguaje de programación interpretado orientado a objetos, de alto nivel con semántica dinámica. Sus estructuras de datos de alto nivel, combinado con su tipado y unificación dinámica lo hacen muy atractivo para el desarrollo rápido de aplicaciones, así como para su uso como un lenguaje

¹⁰ Los scripts son un conjunto de instrucciones generalmente almacenadas en un archivo de texto que deben ser interpretados línea a línea en tiempo real para su ejecución(42).

de scripting o para unificar componentes ya existentes. Cuenta con una sintaxis simple y fácil de aprender, hace hincapié en la legibilidad y, por tanto, reduce el coste de mantenimiento del programa. Admite la incorporación de módulos y paquetes, lo que fomenta la modularidad del programa y la reutilización de código. El intérprete y la extensa biblioteca de estándares con los que cuenta Python están disponibles en formato fuente o binario sin cargo para todas las plataformas, y se pueden distribuir libremente. Es un lenguaje de programación con licencia de código abierto y multiplataforma(44).

Lenguajes de estructuración y diseño de documentos

HTML

HTML (Hypertext Markup Language por sus siglas en inglés) es un lenguaje de marcado para describir la estructura de las páginas web. Brinda a los autores los medios para(45):

- Publicar documentos en línea con encabezados, texto, tablas, listas, fotos, entre otros elementos.
- Recuperar información en línea a través de enlaces de hipertexto.
- Formas de diseño para la realización de transacciones con servicios remotos, para su uso en la búsqueda de información, hacer reservas y pedidos de productos.
- Incluir videoclips, clips de sonido y otras aplicaciones directamente en sus documentos.

CSS

CSS (Cascading Style Sheets por sus siglas en inglés) es un lenguaje utilizado para describir el modo en que son presentadas las páginas web, incluyendo colores, el diseño y las fuentes. Permite adaptar las páginas a diferentes tipos de dispositivos con tamaños de pantalla variables. CSS es independiente de HTML y se puede utilizar con cualquier lenguaje de marcado basado en XML. La separación de HTML y CSS hace que sea más fácil realizar el mantenimiento a los sitios(45). La utilización de CSS para definir el estilo de un sitio web es extremadamente sencillo pues puede ser embebido dentro del HTML de manera rápida y simple(46).

Fundamentación de la selección de los lenguajes

Para la implementación de la herramienta Moux se hace uso de los siguientes lenguajes:

- JavaScript 1.8.5 como lenguaje interpretado del lado del cliente.
- PHP 5.5.3 como lenguaje del lado del servidor.
- HTML 5 como lenguaje de marcado para la estructuración de las páginas web.
- CSS 3 para definir el diseño y el estilo de las páginas web.

Se seleccionan estos lenguajes debido a que por su amplia utilización en el proceso de desarrollo web se encuentran en constante evolución y perfeccionamiento. Cuentan con comunidades de desarrolladores que los respaldan y por tanto es posible acceder a un gran cúmulo de información y tutoriales para el estudio. Además, son soportados por la inmensa mayoría de los navegadores actuales.

1.4.4 Framework de desarrollo

AngularJS

Se decide utilizar AngularJS 1.2.16 para la confección de la aplicación. Este framework de desarrollo está compuesto por un conjunto de librerías de código abierto que resultan de gran utilidad a la hora de crear aplicaciones que se ejecuten del lado del cliente. AngularJS es un proyecto desarrollado por Google, por lo tanto se encuentra en constante perfeccionamiento, además, cuenta con comunidades de desarrolladores cada vez más grande que lo respaldan(47).

Bootstrap

Se utiliza Bootstrap 3.1.1 como framework para realizar el diseño de la aplicación. Esta herramienta es una combinación de HTML, CSS y JavaScript y está diseñada para ayudar a construir componentes de interfaz de usuario totalmente adaptables a todo tipo de pantallas y dispositivos, con soporte para HTML5 y CSS3(48).

1.4.5 Entornos de Desarrollo Integrado

Un IDE (Integrated Development Environment por sus siglas en inglés) es una herramienta que proporciona soporte a la hora de desarrollar aplicaciones de una manera amigable, brindando ayudas visuales en la sintaxis, plantillas, plugins y sencillas opciones para probar y hacer un debug(49). Entre los IDEs de desarrollo más utilizados para la creación de aplicaciones web según Mozilla Developer Network(50) se encuentran NetBeans IDE, Aptana Studio y Eclipse.

NetBeans IDE

NetBeans IDE es un entorno de desarrollo que permite la creación rápida y fácil de aplicaciones web, Java y móviles, así como aplicaciones HTML5 con HTML, JavaScript y CSS. El IDE también proporciona un gran conjunto de herramientas para desarrolladores de PHP y C / C ++. Es gratuito y de código abierto y tiene una gran comunidad de usuarios y desarrolladores de todo el mundo. Entre las características más notables de NetBeans IDE cabe mencionar(51):

- Posee un editor que le permite a los desarrolladores programar código limpio y organizado.
- Soporta varios lenguajes como Java, C/C++, XML y HTML, PHP, Groovy, Javadoc, JavaScript y

JSP.

- Gestión eficiente, fácil y organizada de los proyectos.
- Permite la integración con repositorios de control de versiones tales como Subversion, Mercurial, y Git.
- Facilita el diseño de interfaces GUI para aplicaciones Java SE, HTML5, Java EE, PHP, C/C++, y Java ME mediante el uso de editores y herramientas arrastrar y soltar.
- Es una herramienta multiplataforma y sin restricciones de licencia.

Aptana Studio

Aptana Studio es un IDE completamente equipado para el desarrollo web. Es una herramienta multiplataforma y de código abierto que permite desarrollar y probar aplicaciones web de manera sencilla haciendo uso de una interfaz intuitiva y fácil de usar. Brinda soporte para las últimas tecnologías empleadas por los navegadores tales como HTML5, CSS3, JavaScript, Ruby, Rails, PHP y Python. Algunas de sus características más destacables son(52):

- Puede ser instalado como IDE independiente o bien se puede utilizar como un plugin para Eclipse.
- Incluye asistencia de código para JavaScript, HTML y CSS, un subrayador de JavaScript, HTML y estructura de código CSS.
- Soporta protocolos como FTP, SFTP Y FTPS.
- Posibilidad de personalizar y extender la interfaz de usuario.
- Prevención de errores y notificación de advertencias.
- Permite integración con el sistema de control de versiones Git.

Eclipse

Eclipse es una plataforma de código abierto diseñada desde cero para la construcción de aplicaciones web y de desarrollo de herramientas. Por su diseño, la plataforma no proporciona una gran cantidad de funcionalidades de usuario final por sí misma, su valor se encuentra en que fomenta el desarrollo rápido de características integradas basadas en un modelo de plugin(53).

Eclipse proporciona un modelo común de interfaz de usuario para el trabajo con herramientas. Está diseñado para ejecutarse en múltiples sistemas operativos al tiempo que proporciona una integración robusta con cada uno(53).

Selección del Entorno de Desarrollo Integrado

Para la implementación de la herramienta se hace uso del entorno de desarrollo integrado NetBeans

IDE 8.0 pues cuenta con características que contribuyen a simplificar y agilizar el proceso de desarrollo.

- Es multiplataforma y sin restricciones de licencia.
- Posee un editor que le permite a los desarrolladores programar código limpio y organizado.
- Soporta los lenguajes JavaScript, CSS y HTML.

A lo anteriormente mencionado se le puede agregar que el equipo de desarrollo siente preferencia por esta tecnología pues es una herramienta con la cual ha trabajado previamente y sobre la que ya tiene un cierto dominio basado en su experiencia.

1.4.6 Otras herramientas utilizadas

Servidor web

Se hace uso de Apache v2.4 como servidor web para el desarrollo de la aplicación. Este es un servidor HTTP de código abierto, multiplataforma, flexible y altamente configurable(54).

1.5 Conclusiones del capítulo

Al término de este capítulo se arriba a las siguientes conclusiones:

- La identificación de los conceptos interfaz de usuario, tema web, Moodle, entre otros, contribuyó a un mejor entendimiento del entorno en que se desenvuelve la investigación.
- Tras el estudio de los sistemas homólogos Artisteer, Gantry, Traxos y algunos otros, se concluye que estos no brindan una solución factible al problema planteado en la investigación. Se hace legítima entonces la necesidad de implementar la herramienta Mox para facilitar el proceso de creación de temas web para la plataforma Moodle.
- La selección de los lenguajes, los frameworks y un entorno de desarrollo integrado, así como la selección de la metodología AUP en su variante UCI, contribuirá a que el proceso de desarrollo de la herramienta Mox sea llevado a cabo con calidad.

Capítulo 2: Propuesta de solución

En el presente capítulo se describen las características del sistema a desarrollar. Se hace uso de un modelo conceptual como medio para describir los principales conceptos asociados al dominio del problema y sus relaciones para lograr un mejor entendimiento. Se realiza la captura de los requisitos funcionales y no funcionales del producto y se hace uso de historias de usuario para la descripción de los mismos. Se describen los artefactos generados durante el análisis y diseño de la solución y se definen los patrones arquitectónicos y de diseño a utilizar.

2.1 Descripción de la propuesta de solución

La presente investigación propone como solución una herramienta que facilite la creación de temas web para la versión 2.8 de la plataforma Moodle. La misma será funcional independientemente de la plataforma, pues al ser una aplicación web solo será necesario un navegador para poder utilizarla. La aplicación propiciará la creación de temas web partiendo siempre de un tema cargado por defecto, este podrá ser modificado al gusto del usuario haciendo uso de elementos de diseño incluidos en la herramienta. Los temas generados serán exportados en un archivo comprimido en formato ZIP.

La herramienta podrá ser utilizada abiertamente por cualquier usuario, los cuales contarán con libertades de distribución y modificación del producto.

2.2 Modelo conceptual

Un modelo conceptual es una representación de los principales conceptos asociados a un dominio de problema, identificando las relaciones existentes entre ellos. Los elementos tratados en un modelo conceptual no son componentes del software, sino entidades y objetos de la vida real entre los cuales se establecen relaciones de modo que ayuden a formar una idea general del dominio del problema(55).

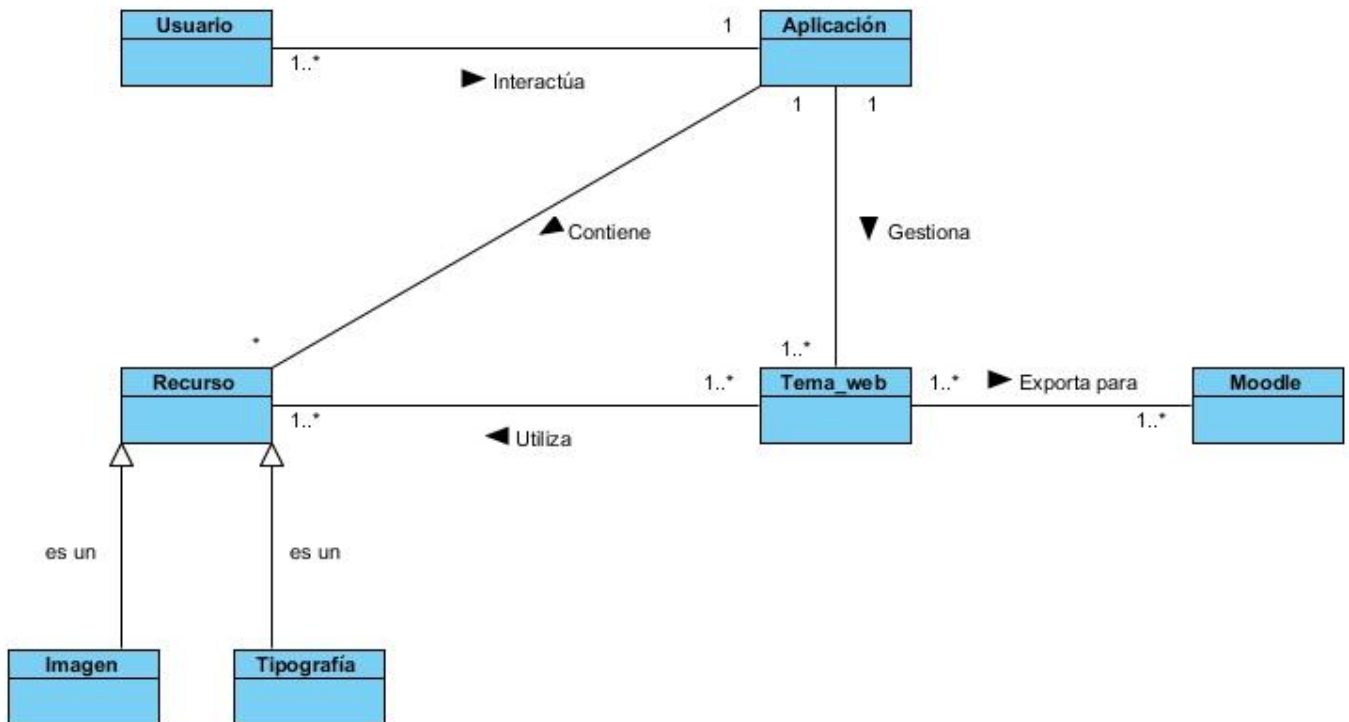


Ilustración 1: Modelo conceptual del dominio

2.2.1 Descripción de los conceptos

Usuario: es la persona que interactúa con el sistema y recopila los recursos y como resultado de esto es generado el tema web.

Aplicación: es el sistema con el cual interactúa el usuario y que contiene un conjunto de recursos. Es utilizado para la creación del tema web y para realizar las operaciones pertinentes sobre el mismo.

Tema web: es el principio bajo el cual el usuario interactúa con el sistema, constituye el resultado final del trabajo del usuario sobre la aplicación que será exportado para su posterior uso en Moodle.

Recurso: está constituido por los elementos que el usuario empleará para la creación del sitio. Pueden ser imágenes y tipografías.

Moodle: es una herramienta para la creación de ambientes de aprendizaje la cual resulta el destino final del tema web, el cual será empleado para modificar la apariencia de esta herramienta.

2.3 Requisitos del sistema

Con el objetivo de garantizar la obtención de una herramienta que posea las características esperadas se procede a definir los requisitos funcionales y no funcionales con los que debe cumplir la solución. Los mismos son enumerados a continuación.

2.3.1 Requisitos funcionales del sistema

Son declaraciones de los servicios que debe proporcionar el sistema. Describen la manera en que el sistema deberá reaccionar ante entradas y situaciones particulares. Los requisitos funcionales de un sistema especifican lo que el sistema debe hacer(56).

Tabla 2 Requisitos funcionales del sistema

Nº	Descripción
RF1	Exportar tema.
RF2	Editar sombra general.
RF3	Editar tipografía general.
RF4	Editar fondo general.
RF5	Editar botones.
RF6	Editar íconos.
RF7	Editar fondo de barra de navegación.
RF8	Editar tipografía de barra de navegación.
RF9	Editar sombra de barra de navegación.
RF10	Editar bordes de barra de navegación.
RF11	Editar estilo de los submenús.
RF12	Editar sombra de los submenús.
RF13	Editar tipografía de los submenús.
RF14	Editar bordes de los submenús.
RF15	Editar fondo de banner.
RF16	Cambiar posición de logo.
RF17	Editar tipografía de banner.
RF18	Editar bordes de banner.
RF19	Editar fondo del deslizador.
RF20	Editar tipografía del deslizador.
RF21	Editar sombra del deslizador.
RF22	Editar enlaces de bloques.
RF23	Editar fondo de bloques.
RF24	Editar tipografía de bloques.
RF25	Editar sombra de bloques.

RF26	Editar bordes de bloques.
RF27	Editar fondo de pie de página.
RF28	Editar tipografía de pie de página.
RF29	Editar enlaces de pie de página.
RF30	Editar bordes de pie de página.
RF31	Cambiar tamaño del banner.
RF32	Cambiar tamaño de página.
RF33	Editar borde de deslizador.
RF34	Editar fondo de página.
RF35	Editar borde de página.

2.3.2 Requisitos no funcionales del sistema

Los requisitos no funcionales, como su nombre sugiere, son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento(56).

Tabla 3 Requisitos no funcionales del sistema

Requisitos no funcionales	
Requisitos de usabilidad - comprensibilidad	Favorecer el trabajo con la herramienta haciendo uso de una interfaz intuitiva y de fácil entendimiento, de modo que su empleo resulte sencillo para usuarios con conocimientos básicos de informática.
Requisitos de portabilidad - portabilidad	Operar en dispositivos que cuenten con un navegador web instalado tal como Internet Explorer, Opera y Mozilla Firefox. Los dispositivos deberán contar con características que les permitan soportar el correcto funcionamiento de estos navegadores.
Requisitos de portabilidad - adaptabilidad	Visualizar correctamente en dispositivos móviles con resoluciones de pantalla de 320x480 en adelante.

2.4 Historias de usuario

Las historias de usuarios constituyen la técnica usada en el desarrollo ágil de software para capturar la descripción de una característica desde el punto de vista del usuario final. Describen lo que este espera del sistema(57).

Las historias de usuario son escritas en presencia del cliente haciendo uso de un lenguaje poco técnico pero comprensible y estarán delimitadas de modo que los programadores puedan implementarlas en unas pocas semanas. Estas podrán ser añadidas, modificadas, eliminadas o reemplazadas en cualquier momento(57).

A continuación se muestran algunas de las historias de usuario del sistema. El resto de ellas pueden ser vistas en el Anexo 1.

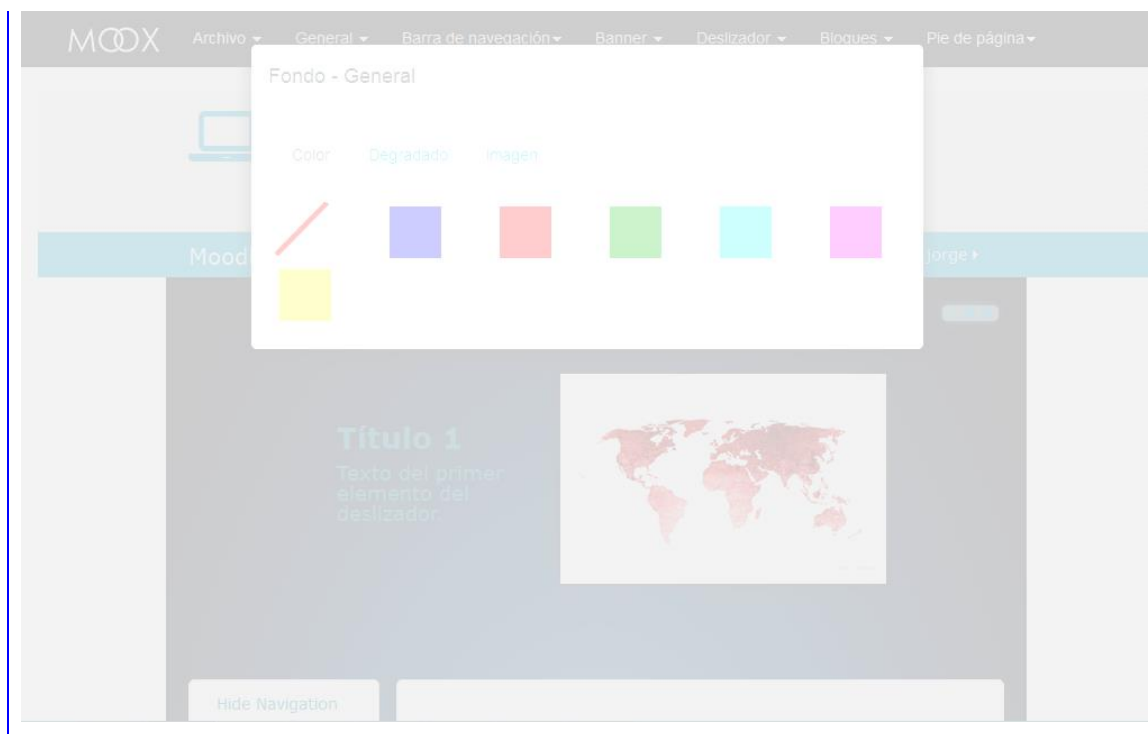
Tabla 4 HU4: Editar fondo general

Número: 4	Nombre del requisito: Editar fondo general
Programador: Jorge Antonio Monedero Soriano	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 1
Riesgo en Desarrollo: N/A	Tiempo Real: 0.1
<p>Descripción:</p> <p>1- Objetivo: Editar el fondo general del tema web.</p> <p>2- Acciones para lograr el objetivo(precondiciones y datos): Se deben tener en cuenta los datos:</p> <ul style="list-style-type: none"> ▪ Color: selector de color ▪ Imagen: botón ▪ Estilo degradado: botón <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> ▪ El usuario selecciona la opción Fondo del menú General. ▪ El sistema muestra una ventana emergente donde brinda al usuario la posibilidad de elegir modificar el fondo general del tema seleccionando un color sólido, un degradado o una imagen. 	

- En caso de que el usuario desee modificar el fondo general aplicando un color sólido accede a la pestaña Color.
 - El sistema muestra varios selectores de color con los colores definidos por defecto.
 - El usuario selecciona un color haciendo uso de los selectores de color.
 - El sistema establece como fondo general el color seleccionado.
- En caso de que el usuario desee modificar el fondo general aplicando una imagen accede a la pestaña Imagen.
 - El sistema muestra las distintas imágenes existentes para utilizar como fondo.
 - El usuario selecciona una de las imágenes mostradas.
 - El sistema establece como fondo general la imagen seleccionada.
- En caso de que el usuario desee modificar el fondo general aplicando un degradado accede a la pestaña Degradado.
 - El sistema muestra 2 selectores de color con los colores definidos por defecto y 3 botones para definir el estilo de degradado.
 - El usuario selecciona los colores que desea utilizar en el degradado así como un estilo.
 - El sistema establece como fondo general un degradado con las opciones seleccionadas.

Observaciones:

Prototipo de interfaz:



2.5 Análisis y diseño de la propuesta de solución

Con el análisis y diseño de la propuesta de solución es posible refinar los requisitos en caso de considerarse necesario, en aras de alcanzar un mayor entendimiento de estos, y una descripción que sea más fácil de mantener y ayude a la estructuración de la solución. Además es modelado el sistema de manera que soporte todos los requisitos, tanto funcionales como no funcionales(32).

2.5.1 Diagrama de clases del análisis

Entre las características del trabajo con el framework AngularJS se puede mencionar la existencia de una clase controladora que albergará las funciones utilizadas en la aplicación. Debido a esto, en la confección de la propuesta de solución se definirá una clase controladora llamada MoxController en la cual estarán contenidos la mayoría de los métodos necesarios para el funcionamiento de la herramienta. Esto trae como consecuencia que durante el proceso de diagramado de las clases del análisis se obtenga un flujo común para todos los requisitos funcionales de la solución, excepto para el **RF1 exportar tema web**, por tanto, se elaboran únicamente dos diagramas de clases del análisis.

A continuación, se muestran ambos diagramas de clases del análisis:



Ilustración 2: Diagrama de clases del análisis



Ilustración 3: Diagrama de clases del análisis (RF1 Exportar tema web)

En los diagramas presentados se aprecian las clases que intervienen en el proceso de creación de los temas. En la ilustración 2 se aprecia la secuencia para realizar la personalización del tema, en esta imagen se muestra como el cliente interactúa con la clase interfaz index seleccionando las opciones de personalización que desea llevar a cabo, seguidamente, esta clase interfaz realiza una llamada a los métodos contenidos en la clase controladora, la cual modifica a la clase entidad Moodle y actualiza inmediatamente la interfaz acorde a los cambios realizados. Por otra parte en la ilustración 3, cuando se procede a exportar el tema web, el usuario interactúa con la clase interfaz, esta notifica a la clase controladora MooxController la cual haciendo uso de otra clase controladora llamada exportar realiza las operaciones necesarias para que el tema sea creado y posteriormente descargado para su uso.

2.5.2 Diagramas de colaboración del análisis

Los diagramas de colaboración ilustran las interacciones existentes entre objetos representadas en un formato de grafo o red, en el cual los objetos pueden ser colocados en cualquier lugar del diagrama(52). Se muestran los diagramas de colaboración correspondientes a la historia de usuario HU4 Editar fondo general. El resto de los diagramas se pueden ver en el Anexo 2.

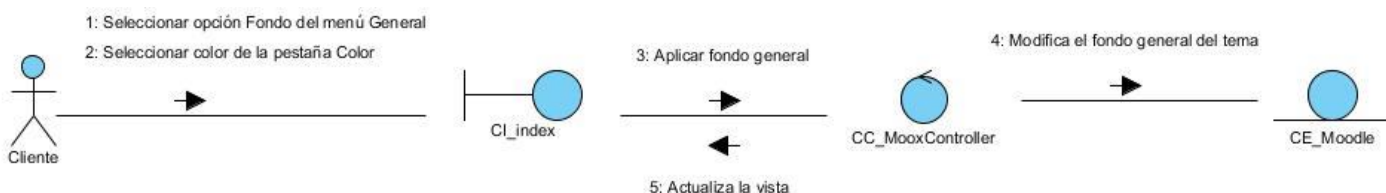


Ilustración 4 DCA_HU4: Editar fondo general - escenario 1



Ilustración 5 DCA_HU4: Editar fondo general - escenario 2

En las imágenes previamente mostradas, es descrito el proceso colaborativo entre las clases del análisis para modificar el fondo general del tema web. El cliente interactúa con la interfaz y selecciona un fondo para el tema, posteriormente, la clase interfaz notifica a la controladora que debe aplicar el fondo de acuerdo a las características elegidas por el usuario. Este efectúa los cambios pertinentes y finalmente actualiza la interfaz.

2.5.3 Arquitectura

La IEEE¹¹ define arquitectura de software como “la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución”(58). La arquitectura de software hace uso de un conjunto de patrones y abstracciones que conforman un hilo conductor que guía el desarrollo del software(58). Es por tal razón que se hace necesario realizar la selección de los patrones arquitectónicos y de diseños a emplear en el desarrollo de la aplicación.

2.5.3.1 Patrones arquitectónicos

Un patrón arquitectónico expresa un esquema de organización estructural fundamental para los sistemas de software. Provee una colección de subsistemas predefinidos, especifica sus responsabilidades, e incluye reglas y líneas guías para organizar las relaciones entre ellos. Los patrones arquitectónicos no son más que una plantilla para arquitecturas de software concretas, que definen las propiedades estructurales de un sistema y que tienen un impacto en la arquitectura de sus subsistemas(59).

La herramienta está desarrollada con el framework JavaScript AngularJS, el cual emplea el patrón arquitectónico MVC (Modelo-Vista-Controlador), aunque no en su versión tradicional sino en una variante ampliamente utilizada en el trabajo con JavaScript denominada MVVM (en español, Modelo-Vista-Modelo de vista). Por tal motivo la aplicación hereda el uso de dicho patrón.

El MVC plantea la separación de las interfaces de usuario, los datos y la lógica de control de la aplicación en tres componentes diferentes: la Vista, el Controlador y el Modelo. El MVVM por su parte, presenta como principal diferencia que se sustituye el Controlador por el Modelo de vista que sincroniza la

¹¹ Institute of Electrical and Electronics Engineers (Instituto de Ingeniería Eléctrica y Electrónica)

información en lugar de hacerlo un Controlador. Este patrón arquitectónico permite la utilización de doble binding, la traducción de “binding” al español sería “enlace” y es precisamente esta la función que realiza, enlazar los datos del modelo con la vista y viceversa. Esto quiere decir que en lugar de controlar manualmente los cambios en la vista o en los datos, estos se actualizan automáticamente cuando se realiza algún cambio en uno de ellos.

El modelo es el responsable de manejar los datos de la aplicación. Responde a las peticiones realizadas por la vista y a las instrucciones del modelo de vista para actualizarse a sí mismo.

La vista está compuesta por todos los elementos visuales de la aplicación. Es decir, es la encargada de mostrar al usuario los medios que este necesita para interactuar con el sistema y la información generada en el modelo como resultado de esta interacción.

El modelo de vista realiza la función de intermediario entre la vista y el modelo, procesa todas las peticiones de la vista hacia el modelo y maneja las reglas del negocio. Además, provee la comunicación con aplicaciones y fuentes externas.



Ilustración 6: Patrón Modelo-Vista-Modelo de vista

El uso de este patrón arquitectónico para el diseño de la herramienta trae como ventajas:

- Se simplifican las labores de mantenimiento y se reduce el tiempo necesario para las mismas.
- Es posible la modificación de las vistas sin necesidad que se paralice todo el sistema.
- Posee soporte para múltiples vistas, debido a que la Vista se separa del Modelo y no hay ninguna dependencia directa entre ambos.

2.5.3.2 Patrones de diseño

Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de

software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular.

A pesar de ser patrones de menor escala que los patrones arquitectónicos, estos suelen ser independientes de lenguajes y paradigmas de programación(59).

Patrones GRASP:

Los Patrones de Asignación de Responsabilidades (GRASP por sus siglas en inglés) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones(55).

En el desarrollo de la solución se emplean los siguientes patrones GRASP:

Creador: el patrón creador guía la asignación de responsabilidades relacionadas con la creación de objetos(55). El uso de este patrón se evidencia en la clase MainController la cual se encarga de la creación de objetos angular.

```
var proyecto=angular.module("Moox", []);
```

Ilustración 7: Ejemplo de uso del patrón creador

Controlador: el patrón controlador asigna la responsabilidad a una clase controladora de realizar las operaciones del sistema como respuesta a los eventos del mismo(55). La utilización de este patrón se aprecia en la clase MainController, la cual maneja los eventos del sistema y realiza las operaciones de edición.

```

$scope.prevFont = ["", "", "", "", ""];

$scope.aplicarTipografia = function (elemento, fuente) {...}

$scope.prevColorTipografia = ["", "#333", "", ""];

$scope.aplicarColorTipografia = function (elemento, color) {...}

$scope.degradado1 = ["#2989cc", "#2989cc", "#2989cc", "#2989cc", "#2989cc", "#2989cc", "#f5f5f5", "#e6e6e6"]
$scope.degradado2 = ["#000000", "#000000", "#000000", "#000000", "#000000", "#000000", "#f5f5f5", "#e6e6e6"]
$scope.tipo_deg = ['h', 'h', 'h', 'h', 'h', 'h', 'h', 'h', 'h', 'h'];
$scope.aplicarFondo = function (elemento, tipo, degradado, src) {...}

$scope.currentSombra = [1, 1, 1, 1, 1];
$scope.currentSombraColor = [null, null, null, null, null];

$scope.aplicarSombra = function (elemento, index, color) {...}

$scope.cambiarEstiloSubmenu = function (index) {...}

```

Ilustración 8: Ejemplo de uso del patrón controlador

2.5.4 Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación(55). Según se explica en el subepígrafe **2.5.1 Diagrama de clases del análisis**, debido a las características del trabajo con AngularJS el proceso de diagramado de clases resulta en solamente dos diagramas para todos los requisitos funcionales. Esto se hace extensivo al diagrama de clases del diseño resultando también en dos diagramas de clases del diseño.

A continuación se muestra el diagrama de clases del diseño de la propuesta de solución:

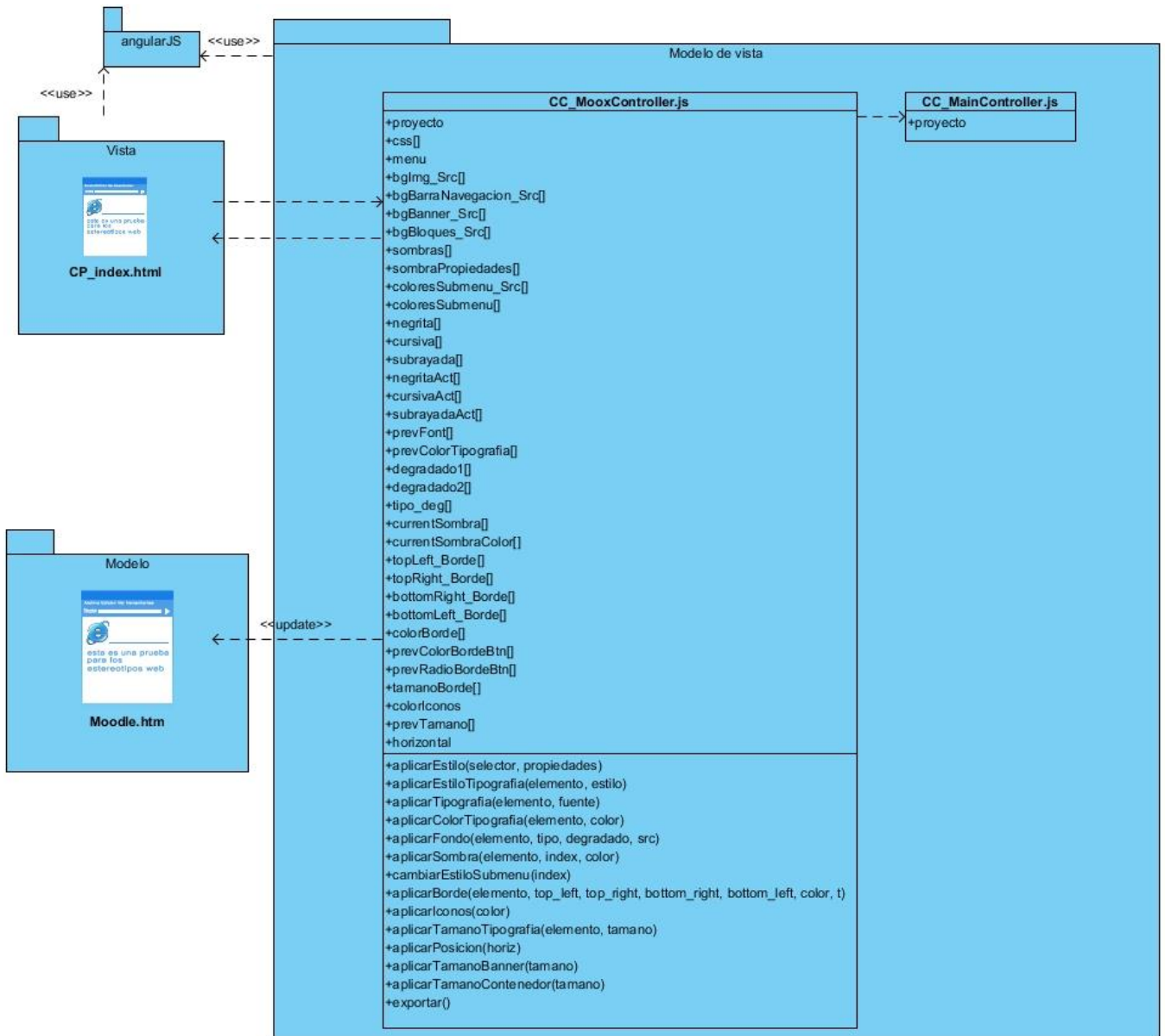


Ilustración 9 Diagrama de clases del diseño

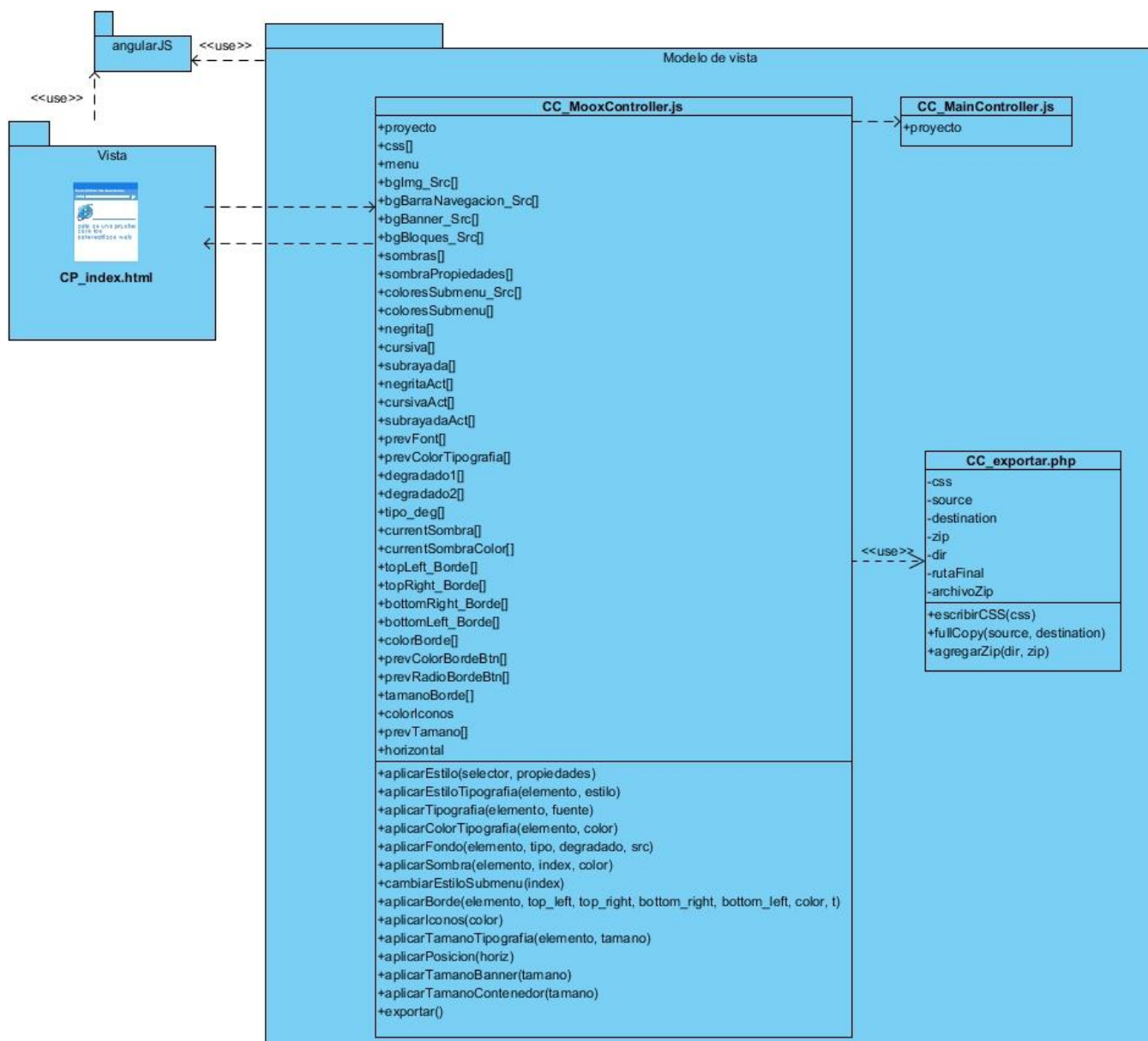


Ilustración 10 Diagrama de clases del diseño (RF1 Exportar tema web)

En las ilustraciones previamente mostradas, es apreciable la relación entre las clases del sistema. Se pueden ver las clases que componen el sistema y que permiten llevar a cabo el proceso de creación de temas web. Se exhiben los métodos y los atributos albergados en las clases controladoras los cuales hacen posible que los temas sean creados y posteriormente exportados.

2.5.5 Diagramas de secuencia del diseño

Los diagramas de secuencia ilustran las interacciones entre objetos con el transcurso del tiempo en un tipo de formato con aspecto de una valla, en el que cada objeto nuevo se añade a la derecha. Estos

muestran los objetos partícipes de la interacción y la secuencia de los mensajes intercambiados(55). A continuación, se muestran los diagramas de secuencia de la historia de usuario HU4: Editar fondo general. El resto de los diagramas pueden ser vistos en el Anexo 3.

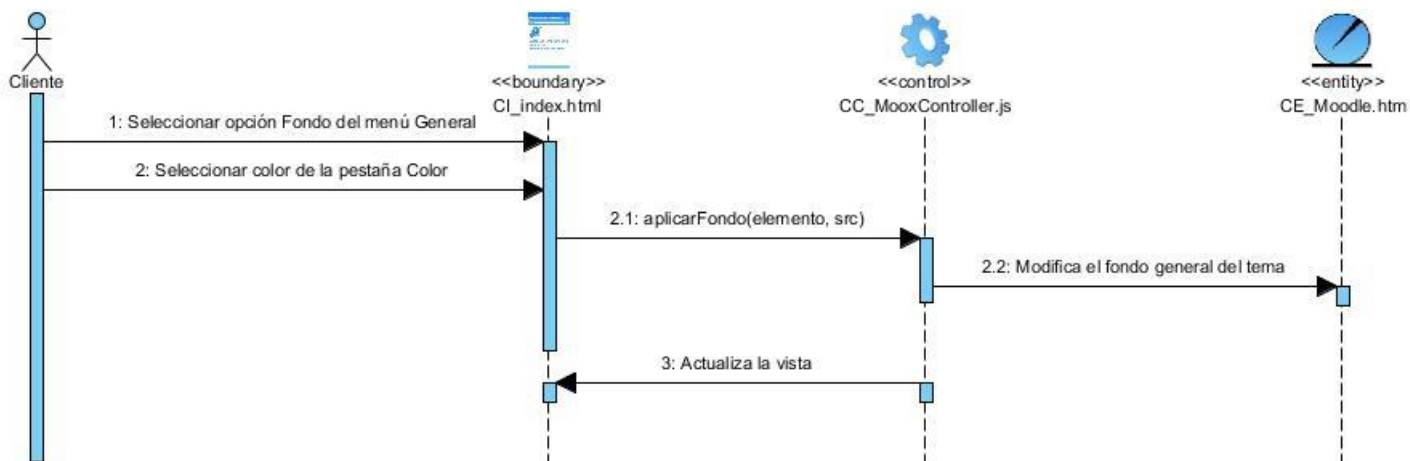


Ilustración 11 DSD_HU4: Editar fondo general – escenario 1

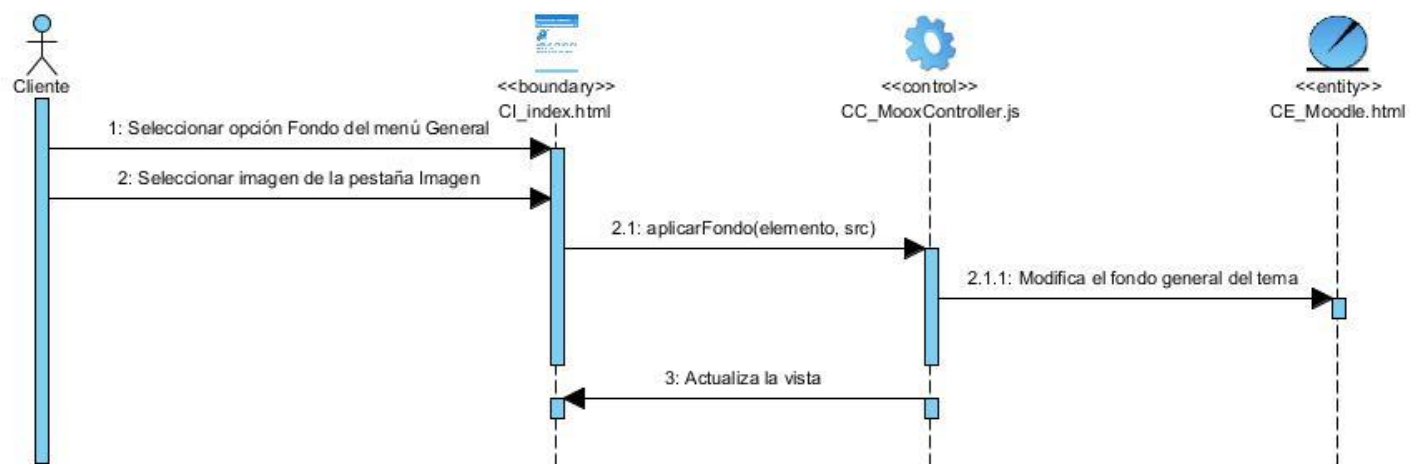


Ilustración 12 DSD_HU4: Editar fondo general – escenario 2

En estos diagramas se muestra el flujo secuencial de las operaciones llevadas a cabo por el sistema para modificar el fondo del tema web. En este caso, este proceso es mostrado de un modo más técnico que en los diagramas de colaboración, donde una vez que el usuario selecciona las opciones necesarias para editar el fondo del tema, index.html hace un llamado al método aplicarFondo() de la clase MooxController.js pasándole los parámetros esperados. Posteriormente esta lleva a cabo la personalización y actualiza la interfaz.

2.5.6 Diagrama de despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware, y permite observar cómo los elementos y artefactos del software se trazan en los mismos(60).

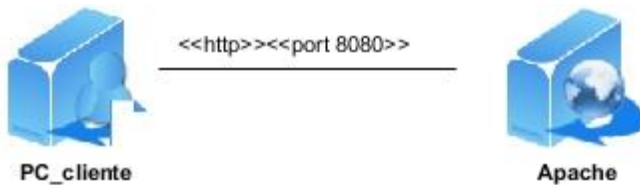


Ilustración 13: Diagrama de despliegue

2.6 Conclusiones del capítulo

Al culminar el capítulo se arriba a las siguientes conclusiones:

- La identificación de los requisitos funcionales y no funcionales con los que debe cumplir la herramienta Moux, y sus descripciones mediante historias de usuario sirvió de guía para la implementación de las distintas funcionalidades de la propuesta de solución.
- La descripción de los requisitos mediante la creación de 35 historias de usuario, sirvió de hilo conductor en el proceso de desarrollo de la solución al quedar plasmada la iteración y el tiempo de desarrollo de cada funcionalidad de la herramienta.
- La selección de los patrones controlador y creador aportó sencillez al proceso de mantenimiento y permitirá la obtención de una herramienta robusta.
- El diagramado de los artefactos generados durante al análisis y diseño de la solución permitió representar las clases del sistema, así como las relaciones existentes entre ellas, de modo que se pudiera apreciar cómo sería el funcionamiento de la herramienta, contribuyendo esto a un mejor entendimiento del sistema durante la implementación.

Capítulo 3: Implementación y pruebas

En el presente capítulo se procede a realizar la implementación y las pruebas de software para garantizar la obtención de un producto de calidad. Se crea el diagrama de componentes del sistema y se definen los estándares de codificación empleados para el desarrollo de la solución. La realización de pruebas a un producto asegura la satisfacción del cliente con el mismo, para lograr esto son definidos los niveles y técnicas de pruebas que le serán aplicadas. Se diseñan también casos de pruebas que servirán para llevar a cabo estas comprobaciones de la calidad.

3.1 Diagrama de componentes

Un diagrama de componentes muestra la división entre los distintos elementos de software que componen un sistema y las relaciones existentes entre los mismos. Permite visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que estos componentes proporcionan y usan a través de interfaces(61).

Seguidamente, se puede observar el diagrama de componentes de la herramienta.

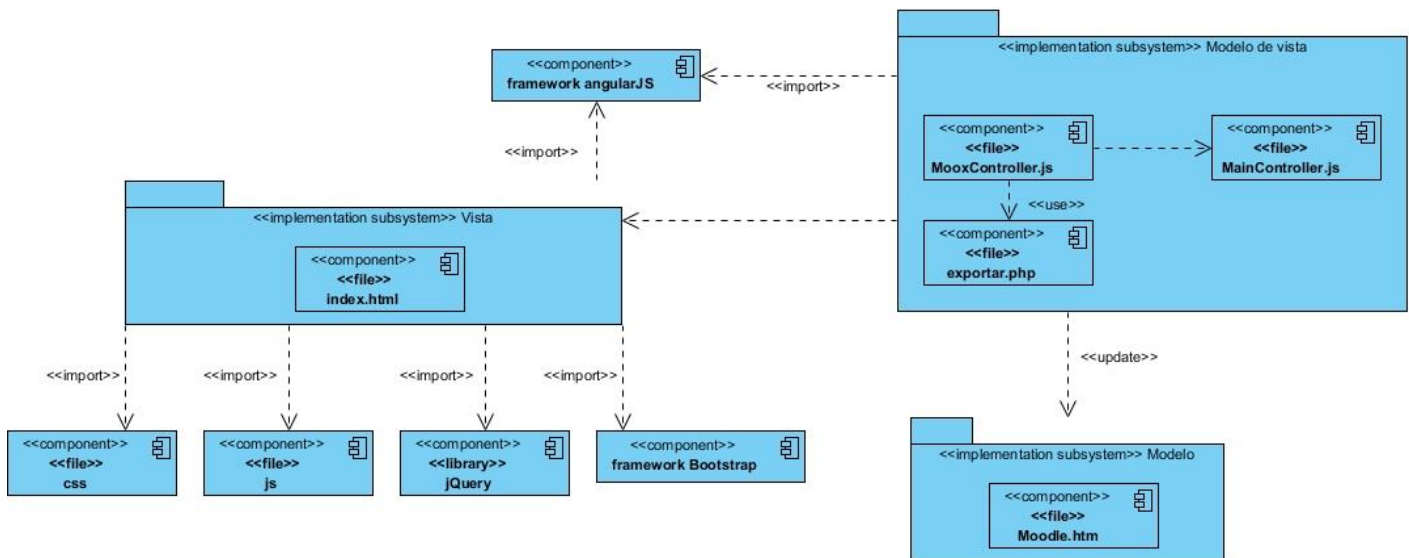


Ilustración 14 Diagrama de componentes del sistema

3.3 Estándares de codificación

Un estándar de codificación es un conjunto de reglas que se utilizan para escribir archivos de código fuente con el objetivo de lograr estructuras de código mucho más comprensibles e identificables para otros programadores diferentes al autor(62). Para el desarrollo de la aplicación se decide utilizar como estándar de codificación las pautas recomendadas por la Guía de estilo JavaScript de Google. A continuación, se describen las pautas adoptadas para la implementación de la propuesta de solución.

Convenciones de nombres

- **Variables**

Las variables tienen que declararse utilizando la forma '*lowerCamelCase*'¹². No deben ir precedidas de un guion bajo para indicar que se tratan de variables privadas.

Ejemplo: `$scope.sombraPropiedades`

- **Métodos**

Los métodos de los objetos tienen que ser nombrados utilizando '*lowerCamelCase*'.

Ejemplo: `$scope.aplicarEstiloTipografia`

3.2 Pruebas de software

El desarrollo de sistemas de software implica la realización de ciertas actividades con el objetivo de detectar y corregir errores introducidos en el producto durante las fases previas de su confección. Estas pruebas son una parte imprescindible en el proceso de desarrollo de software, y están enfocadas a evaluar la calidad del producto(63).

3.2.1 Niveles de prueba

El proceso de realización de pruebas está compuesto por un conjunto de niveles, agrupados teniendo en cuenta sus objetivos y las etapas del desarrollo de software en las cuales son aplicadas(63).

Para validar el correcto funcionamiento de la herramienta, esta será sometida a los siguientes niveles de prueba:

¹² CamelCase es un estilo de escritura que se aplica a frases o palabras compuestas. En el caso de lowerCamelCase la primera palabra se escribe con minúscula y el resto es escrito con mayúscula.

- **Pruebas de sistema**

La prueba de sistema está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas(63).

- **Pruebas de aceptación o validación**

Las pruebas de aceptación son llevadas a cabo para permitir que el cliente valide todos los requisitos. Son realizadas por el usuario final en lugar del responsable del desarrollo del sistema. Una prueba de aceptación puede ir desde la realización de casos de prueba hasta la ejecución de una serie de pruebas bien panificadas. De hecho, la prueba de aceptación puede tener lugar a lo largo de semanas o meses, descubriendo así errores acumulados que pueden ir degradando el sistema(63).

3.2.2 Técnicas de prueba

Las técnicas de prueba facilitan una guía sistemática para diseñar pruebas que comprueben la lógica interna de los componentes de software y verifiquen los dominios de entrada y salida del programa para descubrir errores en la funcionalidad, el comportamiento y el rendimiento(63). Seguidamente se describe en qué consisten las pruebas de caja blanca y caja negra. Téngase en cuenta que para comprobar la calidad del software serán aplicadas pruebas de caja negra.

- **Pruebas de caja blanca**

Las pruebas de caja blanca son un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Haciendo uso de esta técnica se puede lograr que las pruebas diseñadas garanticen que se ejecuten(63):

- Todos los caminos independientes de cada módulo por lo menos una vez.
- Todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Todos los bucles en sus límites y con sus límites operacionales
- Las estructuras internas de datos para asegurar su validez.

▪ **Pruebas de caja negra**

Las pruebas de caja negra se centran en los requisitos funcionales del software. Permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Con las pruebas de caja negra, se pretende demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta. Estas pruebas no pretenden inmiscuirse en el funcionamiento del código, es decir, lo que se comprueba es lo que hace, y no como lo hace. Se espera encontrar con las pruebas de caja negra errores de las siguientes categorías(63):

1. Funciones incorrectas o ausentes.
2. Errores de interfaz.
3. Errores en estructuras de datos o en accesos a bases de datos externas.
4. Errores de rendimiento.
5. Errores de inicialización y de terminación.

3.2.2.1 Casos de prueba

Consiste en el diseño de pruebas que tengan la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y de tiempo. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces(63).

Para probar el correcto funcionamiento de la aplicación se diseñaron un conjunto de casos de prueba que así lo confirmen. El caso de prueba del requisito funcional **Editar fondo general** es mostrado a continuación. Los restantes podrán ser visualizados en el Anexo 4.

Tabla 5 CP Editar fondo general

Descripción							
Permitir la edición del fondo general del tema web.							
Condiciones de ejecución							
El usuario deberá haber accedido a la herramienta previamente.							
Escenario	Descripción	Color	Estilo degradado	Imagen	Respuesta sistema	del	Flujo central
EC 1.1 Opción Editar fondo	Selecciona la opción Fondo del menú General y se muestra una				Brinda la posibilidad de editar el fondo general del tema		General/ Fondo/ Color

<p>general del tema utilizando un color sólido.</p>	<p>ventana emergente que permite al usuario decidir si desea modificar el fondo general utilizando un color sólido o una imagen. Se elige la pestaña Color y se muestran varios selectores de color. Se selecciona uno de ellos y se despliega un selector en el cual se elige el color que se desea utilizar como fondo y se pulsa el botón Elegir. Seguidamente se establece como fondo general del tema el color elegido.</p>			<p>eligiendo un color sólido mediante la utilización de un selector de color. Permite además: cancelar la operación en cualquier momento.</p>	
<p>EC 1.2 Opción Editar fondo general del tema utilizando una imagen.</p>	<p>Selecciona la opción Fondo del menú General y se muestra una ventana emergente que permite al usuario decidir si desea modificar el fondo general utilizando un color sólido o una imagen. Se elige la pestaña Imagen y se muestran varias imágenes. Se selecciona una de ellas y seguidamente se establece como fondo general del tema la imagen elegida.</p>			<p><i>Brinda la posibilidad de editar el fondo general del tema tras seleccionar una de las imágenes mostradas. Permite además: cancelar la operación en cualquier momento.</i></p>	<p>General/ Fondo/ Imagen</p>

<p>EC 1.3 Opción Editar fondo general del tema utilizando un degradado.</p>	<p>1.3 Selecciona la opción Fondo del menú General y se muestra una ventana emergente que permite al usuario decidir si desea modificar el fondo general utilizando un color sólido, un degradado o una imagen. Se elige la pestaña Degradado y se muestran 2 selectores de color y 3 botones para establecer un estilo de degradado. Se seleccionan los colores y el estilo de degradado que se desea emplear. Seguidamente se establece como fondo general del tema el degradado con las características elegidas.</p>				<p>Brinda la posibilidad de editar el fondo general del tema estableciendo un degradado con las características elegidas. Permite además: cancelar la operación en cualquier momento.</p>	<p>General/ Fondo/ Degradado</p>
<p>EC 1.4 Opción de Cerrar.</p>	<p>1.4 Selecciona la opción de Cerrar.</p>				<p>Cierra la ventana.</p>	<p>General/ Fondo/ Cerrar</p>

3.2.3 Resultados de las pruebas de software

La realización de pruebas funcionales permitió identificar un total de 9 no conformidades mostradas a continuación:

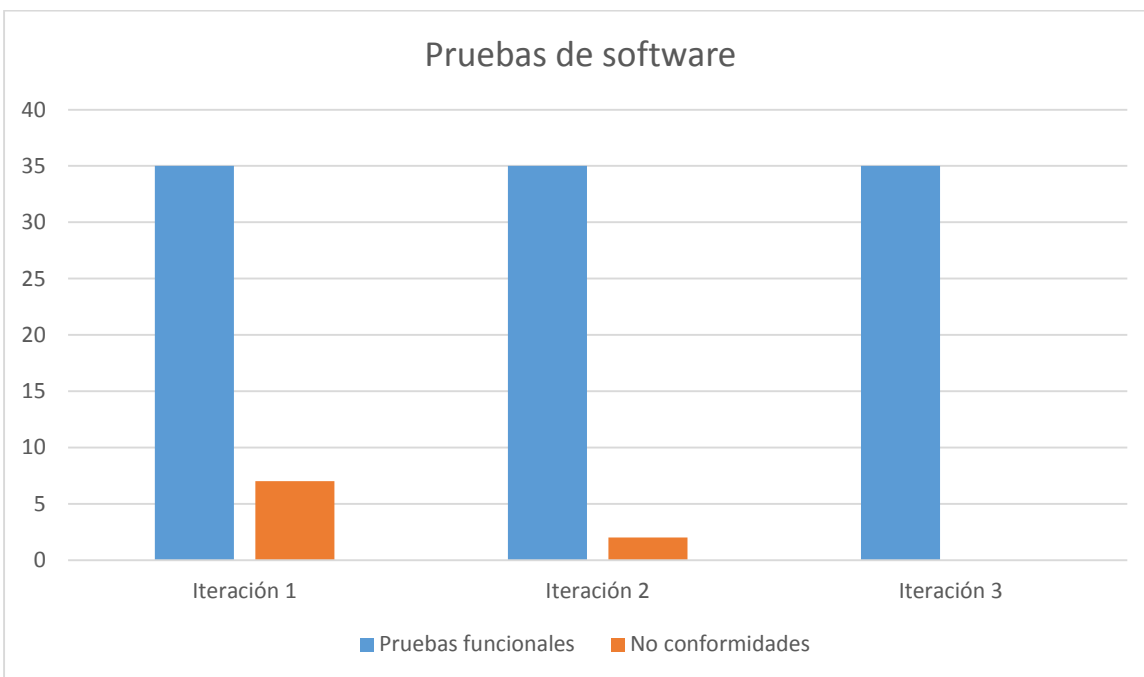
Tabla 6: No conformidades identificadas

No. NC	Requisito Funcional	Descripción	Complejidad	Estado
1	RF 5	No se aplica el estilo a los botones presionados.	MEDIA	Resuelta
2	RF 7	El fondo aplicado a	BAJA	Resuelta

		la barra de navegación no se visualiza.		
3	RF 8	Al editar la tipografía en la barra de navegación también se modifica la tipografía de los submenús.	BAJA	Resuelta
4	RF 22	Al modificar el estilo de los enlaces sobre los cuales se posiciona el cursor también se modifica el estilo de los enlaces en reposo.	BAJA	Resuelta
5	RF 23	No se modifica el fondo de todos los bloques.	BAJA	Resuelta
6	RF 28	Al modificar la tipografía del pie de página se modifica también la tipografía de los enlaces.	BAJA	Resuelta
7	RF 1	Al exportar no se muestra la alerta que informa al usuario el estado del proceso.	BAJA	Resuelta
8	RF32	Cuando se modifica	MEDIA	Resuelta

		el tamaño de la página se superponen las imágenes y el texto del deslizador.		
9	RF32	Cuando se modifica el tamaño de la página pierden la proporción los elementos contenidos en los bloques.	MEDIA	Resuelta

En la siguiente gráfica se muestran los datos correspondientes a las pruebas realizadas en cada iteración:



Se realizaron un total de 3 iteraciones de prueba. En todas las iteraciones fueron aplicados 35 casos de prueba, identificándose 7 no conformidades durante la primera iteración, dichas no conformidades fueron corregidas satisfactoriamente. Posteriormente se realizó la segunda iteración de pruebas donde se identificaron 2 no conformidades de los 35 casos de prueba aplicados. Estas no conformidades fueron

solucionadas y se dio inicio a una última iteración en la cual no se detectaron no conformidades, finalizando de este modo las pruebas funcionales aplicadas a la herramienta.

Al producto también se le realizaron comprobaciones para garantizar que se cumplieran los requisitos no funcionales identificados. Este fue ejecutado en los navegadores Internet Explorer v11, Mozilla Firefox v40.0.2 y Opera v27.0 presentando un correcto desempeño en cada uno de ellos. Además, la aplicación fue cargada en distintos dispositivos móviles con resoluciones de 320x480 en adelante, en cada uno de estos la herramienta se mostró correctamente, adaptándose a cada una de las resoluciones.

3.3 Conclusiones del capítulo

- La creación de un diagrama de componentes permitió realizar un desglose de los diferentes elementos de software que componen la herramienta para un mayor entendimiento de cómo está conformada.
- Tras definir como estándar de codificación las pautas recomendadas por la guía de estilo JavaScript de Google se obtiene un código fuente limpio que asegure un fácil entendimiento y futuro mantenimiento del mismo.
- Tras el diseño de 35 casos de prueba fueron aplicadas las pruebas funcionales y de aceptación, las cuales permitieron identificar las no conformidades que presentaba la herramienta Mox, posibilitando su corrección para obtener finalmente un producto que cumple con los requisitos especificados.

Conclusiones generales

Al término del presente trabajo de diploma se ha dado cumplimiento al objetivo general y a los objetivos específicos definidos en la investigación y se arriba a las siguientes conclusiones:

- Con la investigación realizada se determinó la necesidad de confeccionar la herramienta Mox para la creación de temas web para la plataforma Moodle, y permitió la selección de la metodología AUP en su variante UCI para guiar el proceso de trabajo, así como las herramientas y tecnologías utilizadas para la elaboración de la solución.
- Con el análisis y diseño de la herramienta se obtuvieron un conjunto de artefactos, los cuales propiciaron un mejor entendimiento del funcionamiento de Mox y de cómo estaría estructurada, favoreciendo la simplicidad de la implementación de las funcionalidades.
- Tras identificar las no conformidades que presentaba la herramienta mediante pruebas funcionales y de aceptación efectuadas, estas fueron corregidas, mejorando así la calidad de la solución.

Recomendaciones

Con el objetivo de ampliar las funcionalidades del sistema de modo que se mantenga acorde a las tendencias y tecnologías más actuales, se proponen las siguientes recomendaciones:

- Ampliar su soporte para otros sistemas gestores de contenido como WordPress, Joomla y Drupal.
- Dar la posibilidad al usuario de incorporar imágenes externas a la aplicación para la creación de los temas.
- Incorporar en la herramienta plantillas prediseñadas para que el usuario pueda elegir y modificar al gusto.

Bibliografía

1. CALLE, Franklin. *Recursos Web 2.0 con Moodle y Google Apps para optimizar el proceso de enseñanza-aprendizaje*. 13 November 2009.
2. GUÀRDIA, Francesc Xavier Ribes. La Web 2.0. El valor de los metadatos y de la inteligencia colectiva. *Telos: Cuadernos de comunicación e innovación*. 2007. No. 73, p. 36–43.
3. Interfaz gráfica de usuario (GUI). [online]. 18 November 2015. [Accessed 18 November 2015]. Available from: <http://www.fundeu.es/escribireninternet/interfaz-grafica-de-usuario-gui/>
4. CAÑELLAS, Alicia. CMS, LMS y CLMS. Definición y diferencias.
5. CORRAL, Yolanda. LMS. Learning Management System. .
6. Educación virtual o educación en línea - Ministerio de Educación Nacional de Colombia. [online]. [Accessed 6 June 2016]. Available from: <http://www.mineducacion.gov.co/1759/w3-article-196492.html>
7. Misión | Portal de la Universidad de las Ciencias Informáticas. [online]. [Accessed 8 February 2016]. Available from: <http://www.uci.cu/?q=mision>
8. Producción. [online]. [Accessed 8 February 2016]. Available from: <https://octavitos.uci.cu/?q=node/69>
9. La enseñanza on-line. [online]. [Accessed 6 June 2016]. Available from: http://www.uhu.es/cine.educacion/didactica/0017ensenanza_online.htm
10. LEWIS, Clayton and RIEMAN, John. Task-centered user interface design. *A Practical Introductio*. 1993.
11. Interfaz de Usuario. *About.com en Español* [online]. [Accessed 2 December 2015]. Available from: <http://computadoras.mac.about.com/od/nuevos-usuarios-mac/g/Interfaz-De-Usuario.htm> Definición del término Interfaz de Usuario.
12. RAMOS, Juan Carlos and VERA, Marcela. *Introducción al Diseño de Sistemas de Información*.
13. Interfaces gráficas de usuario. [online]. [Accessed 24 November 2015]. Available from: <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Interfaces/enlightment/>
14. SHREVES, Ric. *Drupal 7 Themes*. Packt Publishing Ltd, 2011. ISBN 1-84951-277-9.
15. Acerca de Moodle - MoodleDocs. [online]. [Accessed 24 November 2015]. Available from: https://docs.moodle.org/all/es/Acerca_de_Moodle
16. ¿Qué es TemplateToaster. *TemplateToaster* [online]. 11 December 2015. [Accessed 11 December 2015]. Available from: <http://templatetoaster.com/>.
17. Artisteer - web design software and joomla template maker. [online]. [Accessed 3 February 2016]. Available from: <http://www.artisteer.com/>

18. WordPress Theme Generator « Create unlimited wordpress themes with the best generator ever Wordpress Theme Generator. [online]. 11 December 2015. [Accessed 11 December 2015]. Available from: <http://www.wpthemegenerator.com/>
19. Gantry - Next Generation Theme Framework | Gantry. [online]. 10 December 2015. [Accessed 10 December 2015]. Available from: <http://gantry.org/>
20. Homepage - Twig - The flexible, fast, and secure PHP template engine. [online]. [Accessed 3 February 2016]. Available from: <http://twig.sensiolabs.org/>
21. HERNÁNDEZ, Elvis. *Herramienta generadora de plantillas HTML*. Universidad de las Ciencias Informáticas, [no date].
22. TORRES, Alberto. *Herramienta multiplataforma para facilitar el proceso de creación de temas web para los sistemas gestores de contenido*. Universidad de las Ciencias Informáticas, [no date].
23. LABORATORIO NACIONAL DE CALIDAD DEL SOFTWARE. *INGENIERÍA DEL SOFTWARE: METODOLOGÍAS Y CICLOS DE VIDA*. March 2009. INTECO.
24. TICONA, Shirley. *Metodologías tradicionales, metodologías ágiles, metodologías para juegos, metodologías educativas y metodologías para aplicaciones móviles*.
25. FIGUEROA, Roberth G., SOLÍS, Camilo J. and CABRERA, Armando A. *Metodologías Tradicionales vs. Metodologías Ágiles*. *Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación. (En línea), Disponible en: <http://adonisnet.files.wordpress.com/2008/06/articulo-metodologia-de-sw-formato.doc>*. 2008.
26. LETELIER, Patricio and PENADÉS, María Carmen. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. . 2006.
27. BECK, Kent. *Extreme programming explained: embrace change*. Addison-Wesley Professional, 2000. ISBN 0-201-61641-6.
28. Qué es SCRUM. *Proyectos Ágiles* [online]. [Accessed 30 November 2015]. Available from: <http://proyectosagiles.org/que-es-scrum/>
29. PÉREZ PÉREZ, María J. *Guía comparativa de metodologías ágiles*. . 2012.
30. Kanban. *Crisp - Get agile with Crisp* [online]. 30 November 2015. [Accessed 30 November 2015]. Available from: <https://www.crisp.se/gratis-material-och-guider/>
31. EDEKI, Charles. *Agile Unified Process*. *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE*. 2013. Vol. 1, no. 3.
32. RODRÍGUEZ, Tamara. *Metodología de desarrollo para la actividad productiva de la UCI*.
33. Pressman Software Engineering Resources: Software Tools (CASE). [online]. [Accessed 10 December 2015]. Available from: <http://www.rspa.com/spi/case.html>

34. Visual Paradigm frequently asked questions. [online]. [Accessed 1 December 2015]. Available from: <http://www.visual-paradigm.com/support/faq.jsp>
35. 10 reasons to choose Visual Paradigm products. [online]. [Accessed 1 December 2015]. Available from: <http://www.visual-paradigm.com/aboutus/10reasons.jsp>
36. IBM - Rational Rose Enterprise. [online]. 1 December 2015. [Accessed 1 December 2015]. Available from: <http://www-03.ibm.com/software/products/es/enterpriseRational>
37. argouml-stats: Chapter 1. Introduction. [online]. [Accessed 1 December 2015]. Available from: <http://argouml-stats.tigris.org/documentation/quickguide-0.34/ch01.html#d0e117>
38. HERNÁNDEZ, Enrique. *El Lenguaje Unificado de Modelado (UML)*.
39. FRESNO, Víctor. *Lenguajes de marcado. Ofimática avanzada*.
40. TIOBE Software: Tiobe Index. [online]. [Accessed 10 December 2015]. Available from: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
41. ¿Qué es Javascript? *Maestros del Web* [online]. [Accessed 4 December 2015]. Available from: <http://www.maestrosdelweb.com/que-es-javascript/>
42. Definicion de script. [online]. [Accessed 7 December 2015]. Available from: <http://www.alegsa.com.ar/Dic/script.php>
43. PHP: What is PHP? - Manual. [online]. [Accessed 4 December 2015]. Available from: <http://php.net/manual/en/intro-what-is.php>
44. Welcome to Python.org. *Python.org* [online]. [Accessed 7 December 2015]. Available from: <https://www.python.org/doc/essays/blurb/>
45. HTML & CSS - W3C. [online]. [Accessed 7 December 2015]. Available from: <http://www.w3.org/standards/webdesign/htmlcss>
46. Why use CSS? *Mozilla Developer Network* [online]. [Accessed 5 December 2015]. Available from: https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_started/
47. Introduction to AngularJS. [online]. [Accessed 30 May 2016]. Available from: http://www.w3schools.com/angular/angular_intro.asp
48. Getting started · Bootstrap 3.1.1 Documentation - BootstrapDocs. [online]. [Accessed 30 May 2016]. Available from: <http://bootstrapdocs.com/v3.1.1/docs/getting-started/>
49. MENTORING, Global. ¿Qué es un IDE? Global Mentoring - Cursos Java Online | . [online]. [Accessed 8 December 2015]. Available from: <http://globalmentoring.com.mx/cursos-java/java-fundamentos/que-es-un-ide/>

50. JavaScript. *Mozilla Developer Network* [online]. [Accessed 3 December 2015]. Available from: <https://developer.mozilla.org/es/docs/Web/JavaScript>
51. NetBeans IDE - Overview. [online]. [Accessed 3 December 2015]. Available from: <https://netbeans.org/features/index.html>
52. Aptana | Studio. [online]. [Accessed 7 December 2015]. Available from: <http://www.aptana.com/products/studio3.html>
53. Help - Eclipse Platform. [online]. [Accessed 7 December 2015]. Available from: http://help.eclipse.org/mars/index.jsp?topic=%2Forg.eclipse.platform.doc.isv%2Fguide%2Fint_eclipse.htm
54. FAQ - Httpd Wiki. [online]. [Accessed 3 May 2016]. Available from: http://wiki.apache.org/httpd/FAQ#What_is_Apache.3F
55. LARMAN, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. 1999. Prentice Hall.
56. Requerimientos Funcionales y No Funcionales (RF/RNF). [online]. [Accessed 15 March 2016]. Available from: <http://ingenieriadesoftware.bligoo.com.mx/requerimientos-funcionales-y-no-funcionales-rf-rnf>
57. What is user story? - Definition from WhatIs.com. [online]. [Accessed 16 March 2016]. Available from: <http://searchsoftwarequality.techtarget.com/definition/user-story>
58. HILLIARD, Rich. Ieee-std-1471-2000 recommended practice for architectural description of software-intensive systems. *IEEE*, <http://standards.ieee.org>. 2000. Vol. 12, p. 16–20.
59. BUSCHMANN, Frank, MEUNIER, Regine, ROHNERT, Hans, SOMMERLAD, Peter and STAL, Michael. *A system of patterns: Pattern-oriented software architecture*. . 1996.
60. Sparx Systems - Tutorial UML 2 - Diagrama de Despliegue. [online]. [Accessed 21 April 2016]. Available from: http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html
61. Diagramas de componentes de UML: Referencia. [online]. [Accessed 22 April 2016]. Available from: <https://msdn.microsoft.com/es-es/library/dd409390.aspx>
62. BETANCUR, Alejandro Villa and PLAZA, Jorge E. Giraldo. Automatización de pruebas unitarias de códigos PHP. *Scientia et Technica*. 2012. Vol. 2, no. 50, p. 147–151.
63. PRESSMAN, Roger. *Ingeniería del Software. Un enfoque práctico*. 5ta. [no date].