



UCI

Universidad de las Ciencias Informáticas

Facultad 4

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

*Módulo de gestión de temas de interfaz de usuario para el
subsistema Bachiller de la Plataforma Educativa Zera*

Autores

Adia Herrera Santos
Lisandra Viamontes Rodríguez

Tutores

MSc. Yaillet Martínez Pérez
Ing. Yerandy Manso Guerra

La Habana, junio 2013

“Año 55 de la Revolución”



"...aquí está una de las tareas de la juventud: empujar, dirigir con el ejemplo la producción del hombre de mañana. Y en esta producción, en esta dirección, está comprendida la producción de sí mismos..."

Ernesto Che Guevara



Declaración de autoría

Declaramos que somos los autores de este trabajo y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas, así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del año 2013.

Autor: Lisandra Viamontes Rodríguez

Autor: Adia Herrera Santo

Tutor: Yaillet Martínez Pérez

Tutor: Yerandy Manso Guerra



Agradecimientos

A mi mamá por todos los sacrificios que ha hecho, porque algún día espero darle aunque sea, la mitad de todo lo que me ha dado.

A mi papá que me han enseñado a perseguir mis metas y sueños hasta lograrlos, por ser mi ejemplo.

A mi madrina por ser mi segunda madre, una madre que solo le faltó parirme, y que me ha dado siempre lo mejor y mi padrino por darme sus consejos y ayudarme a ser mejor persona.

A mi hermana que siempre a estado a mi lado y es "la hermana que más quiero".

A mi abuelo por todos sus consejos y todas las historias, cada una de estas las recordaré por siempre.

A toda mi familia por la preocupación y apoyo durante todo este tiempo.

A mis vecinos que siempre han estado pendiente de la futura ingeniera.

A mis compañeras de cuarto, de clases, de salidas, de paseos, de campismos, etc por ser mis amigas estos cinco años. Por aconsejarme, por estar a mi lado en las buenas y en las malas Anay, Aleyda, Ivis, Anabel, Lianet y los muchachos del segundo piso que siempre han jugado dominó conmigo, por muy mala que sea, por soportarme estos cinco años Basulto, Alexei, Franklin, Yadrían, Juan Carlos y Victor.

A Chi que desde que entré en la universidad me ha acompañado, ella podrá decir otras cosas, pero siempre la consideraré una amiga.

A Liliana por todas la veces que dije hasta aquí, y me dijo "aún te queda camino". Por ayudarme a levantarme una y otra vez. Te agradezco en el alma poder tenerte como amiga.

A Nesty por estar siempre a mi lado cuando te necesito, por quererme y comprenderme, por ser mi amigo incondicional y futuro padrino de mis hijos.

A Wilfredo y Pedro por haberme ayudado con todos mis enredos informáticos.

A Yoandy gracias, por todos los consejos y ayuda que he recibido de ti.

A mis tutores Yerandy y Yaillet gracias por el apoyo durante el desarrollo de esta tesis.

A mi compañera Lisandra por todo su esfuerzo y dedicación a esta tesis.

Adia



Agradecimientos

A mis padres, por creer en mi y sacarme adelante, dándome ejemplos dignos de superación y entrega, porque en gran parte gracias a ustedes, hoy puedo ver alcanzada mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera, y porque el orgullo que sienten por mi fue lo que me hizo ir hasta el final. Va por ustedes, por lo que valen, porque admiro su fortaleza y por lo que han hecho de mí.

A mi hermana Isa, que con su amor me ha enseñado a salir adelante. Gracias por la paciencia, gracias por la preocupación que siempre has tenido y sobre todo, gracias por apoyarme en otro momento tan importante en mi vida.

A mi novio Dennier, mi titi, quien lloró y rió en cada momento junto a mí y fue capaz de contenerme cuando todo iba mal. Gracias por amarme como solo tú lo puedes hacer.

A mis amistades Luis y Meybel por sus consejos y por compartir junto a nosotros muchos momentos agradables.

A mis compañeras Orisel, Aimara, Zaylí, Lizandra, Chi, Camejo y Susana por pasar a mi lado los momentos de mi vida universitaria y estar siempre en las buenas y en las malas. Ya otras que no pudieron llegar hasta el final, jamás lo olvidaré.

A mi compañera de tesis Adia que sin su ayuda este maravilloso momento no hubiera sido posible.

A mis tutores Yailet y Yerandy, por su ayuda, consejos y sugerencias en todo momento.

A La Revolución Cubana por hacer realidad el sueño de convertirme en una profesional y a la Universidad de las Ciencias Informáticas (UCI) por los momentos vividos.

Lisandra



Dedicatoria

Adia:

A mis padres Mirtha y Jesús, por ser mi apoyo incondicional. A mis padrinos Maritza y Germán por ser mis guías. A mi hermana Ania y mi abuelo Pepe porque los quiero. A mis abuelas Mirtha y Aida que aunque estuvieron muy lejos de verme graduar, siempre las tengo presente.

Lisandra

Este maravilloso logro se lo dedico a:

Mis padres por enseñarme a luchar, por su gran corazón y capacidad de entrega, pero sobre todo por enseñarme a ser responsable, gracias a ustedes he llegado a esta meta.

Mi querida hermana por su cariño y apoyo, por darme la fortaleza y motivo para seguir superándome.

Mi novio Dey que con su apoyo constante y amor incondicional ha sido amigo y compañero inseparable durante toda la carrera. Gracias por estar siempre a mi lado.

Mis compañeros por regalarme su amistad a lo largo de estos cinco años.

A todos, espero no defraudarlos y contar siempre con su valioso apoyo, sincero e incondicional.



Resumen

El diseño de interfaz de usuario es una parte fundamental dentro del proceso de desarrollo de un software, influyendo en gran medida en su efectividad comunicacional, facilidad de uso e impacto generado en el usuario. Si se trata de software educativo, es más importante aún el papel de la interfaz, pues además de influir en la aceptación o rechazo del software, puede provocar en el estudiante un rechazo por el tema tratado.

El presente trabajo propone el desarrollo de un módulo de diseño de interfaz de usuario con funcionalidades que faciliten el trabajo de los administradores y la visualización adecuada de los contenidos por parte de los docentes. Se realiza a través de la metodología RUP el análisis y diseño de las características del sistema y posteriormente la implementación y prueba de sus funcionalidades. Como resultado se obtiene un módulo para el subsistema Bachiller, que facilita el proceso de gestión de temas de interfaz de usuario desde la Plataforma Educativa Zera.

Palabras claves: gestión de temas, interfaz de usuario, módulo, plataforma educativa.



Índice de Contenido

Introducción	1
Capítulo I: Fundamentación teórica de la gestión de temas	2
Introducción	2
1.1 Principales conceptos de la investigación	2
1.1.1 Interfaz	2
1.1.2 Interfaz de usuario	2
1.1.3 Ventajas de una apropiada interfaz gráfica de usuario.....	4
1.1.4 Temas de interfaz de usuario.....	5
1.1.5 Diseño de interfaz de usuario en plataformas educativas	6
1.1.6 Estudio de sistemas similares que realizan la gestión de temas	7
1.2 Ambiente de desarrollo	12
1.2.1 Metodología de desarrollo de software	12
1.2.2 Lenguaje de modelado	13
1.2.3 Lenguajes de desarrollo.....	14
1.2.4 Frameworks.....	16
1.2.5 Servidor web.....	19
1.2.6 Entorno de desarrollo	20
1.2.7 Herramientas para el modelado del sistema	21
1.2.8 Herramienta de prototipado	22
1.2.9 Analizador de archivos CSS	22
Conclusiones del capítulo	23
Capítulo II: Características del sistema	24
Introducción	24



2.1	Modelo de Dominio	24
2.1.1	Descripción de los conceptos del dominio	25
2.2	Descripción del sistema propuesto.....	26
2.3	Especificación de los requerimientos del software.....	27
2.3.1	Requerimientos funcionales.....	28
2.3.2	Requerimientos no funcionales.....	30
2.4	Presentación del diagrama de caso de uso.....	32
2.4.1	Descripción del actor del sistema	32
2.4.2	Patrones de casos de uso	32
2.4.3	Diagrama de caso de uso del sistema	33
2.4.4	Descripción de los casos de uso.....	34
	Conclusiones del capítulo	35
	Capítulo III: Análisis y diseño de la propuesta del sistema.....	36
	Introducción	36
3.1	Modelo del análisis.....	36
3.1.1	Diagrama de clases del análisis.....	36
3.2	Patrones arquitectónicos y de diseño en Symfony	37
3.2.1	Patrón arquitectónico Modelo-Vista-Controlador.....	37
3.2.2	Patrones de diseño en Symfony	38
3.3	Modelo de diseño.....	40
3.3.1	Diagrama de clases del diseño	40
3.4	Modelos de datos.....	40
	Conclusiones del capítulo	41
	Capítulo IV: Implementación y pruebas de las funcionalidades.....	42



Introducción	42
4.1 Estándares de codificación.....	42
4.2 Flujo de trabajo de implementación.....	44
4.2.1 Diagrama de componentes.....	44
4.2.2 Diagrama de despliegue.....	45
4.3 Pruebas de software	46
4.3.1 Niveles de Prueba	46
4.3.2 Métodos de Prueba	47
4.3.3 Resultados obtenidos	50
Conclusiones del capítulo	51
Conclusiones	52
Recomendaciones	53
Referencias bibliográficas.....	54
Bibliografía.....	¡Error! Marcador no definido.
Glosario de Términos.....	61
Anexos.....	62



Índice de figuras

Figura 1. Modelo de dominio.....	25
Figura 2. Estructura de los temas	27
Figura 3. Diagrama de casos de uso.....	34
Figura 4. DCA Crear temas de interfaz de usuario.....	36
Figura 5. DCD Crear temas de interfaz de usuario.....	40
Figura 6. Diagrama de Componente general	45
Figura 7. Diagrama de despliegue	46

Índice de tablas

Tabla 1. Descripción del actor del sistema.....	32
Tabla 2. Descripción del CU Crear temas de interfaz de usuario	34
Tabla 3. CP Listar temas de interfaz de usuario.....	48
Tabla 4. Resultados de las pruebas (1era Iteración)	50
Tabla 5. Resultados de las pruebas (2da Iteración)	51



Introducción

El desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) en los últimos años ha motivado a la interacción entre el hombre y los ordenadores. Convirtiéndose en una disciplina que permite el intercambio entre software y personas, donde su objetivo principal es minimizar errores, incrementar la satisfacción, disminuir la frustración y hacer más productivas las tareas que rodean a las personas y los computadores (1).

La web es el principal canal de comunicación y ha pasado de ser un espacio estático, donde solo se consumen contenidos de forma unidireccional, a convertirse en un entorno bidireccional. Este entorno lo permite la segunda generación del desarrollo de tecnología web, la Web 2.0 (2). Esta se basa en que la organización y el flujo de información dependen del comportamiento de las personas que acceden a ella, permitiéndose no sólo un acceso mucho más fácil y centralizado a los contenidos, sino su propia participación tanto en la clasificación de los mismos como en su construcción, mediante herramientas cada vez más fáciles e intuitivas de usar.

Cuando un usuario utiliza una herramienta, o accede e interactúa con un sistema informático, suele existir "algo" entre el usuario y el objeto de interacción. Ese "algo", que es a la vez un límite y un espacio común entre ambas partes, es la interfaz de usuario (UI, por sus siglas en inglés). Hoy en día es posible la interacción con el mundo informático a través de ellas. La interfaz de un programa es un conjunto de elementos hardware y software de una computadora que muestran información al usuario y le permiten interactuar con dicha información y con la máquina (3). Al pensar en la UI la mayoría de los desarrolladores de software piensan en ventanas, menús, gráficos, colores y se olvidan del componente principal: el usuario.

Una de las partes más importantes del software educativo es la UI, pues es la forma mediante la cual el estudiante va a dar instrucciones o va a recibir información, además de influir en la aceptación o rechazo del software. Una interfaz mal concebida puede provocar en el estudiante un rechazo por el tema o materia que se pretende abordar, por lo que debe ser de algún modo sencilla de utilizar para que no constituya una barrera entre el conocimiento a transmitir y la forma de interactuar con el software.

En Cuba la educación progresa con los avances tecnológicos y el uso de plataformas educativas como complemento del proceso de enseñanza-aprendizaje. En la Universidad de las Ciencias Informáticas (UCI)



específicamente el Centro de Tecnologías para la Formación (FORTES), cuenta con varios productos que se encargan de desarrollar tecnologías que permiten ofrecer servicios y productos para la implementación de soluciones de formación, que combinan los elementos pedagógicos y tecnológicos más avanzados, integrando así los procesos de formación, producción e investigación (4). Uno de estos productos es la Plataforma Educativa Zera, compuesta por subsistemas encargados de permitir el control sobre cada una de sus áreas, entre ellos se encuentra el subsistema Bachiller el cual es la base principal de la actividad académica y todo lo gestionado a través del resto de los subsistemas confluye en este. El diseño de la interfaz de usuario de Bachiller está basado en temas predefinidos que identifican el contenido mostrado y el espacio que presenta el subsistema Materia encargado de la gestión de los temas, solo cuenta con las funcionalidades: descargar tema base, cargar los temas creados externamente y listar los temas luego de ser cargados. De esta forma se presentan deficiencias en cuanto a la gestión de temas pues no permite:

- ✓ Crear y modificar los temas de interfaz de usuario desde la plataforma, dificultando el trabajo de los administradores en cuanto a la gestión de los temas.
- ✓ Mostrar una vista previa de los temas, trayendo como consecuencia que puedan existir errores en el diseño de las plantillas.
- ✓ Eliminar los temas que ya no cumplen con los requisitos de los usuarios, lo que dificulta el trabajo de los administradores en cuanto a la posibilidad de eliminar los temas desde la plataforma.
- ✓ Posicionar las pequeñas aplicaciones que contienen los temas, lo que afectaría su adecuada presentación, si se realizan modificaciones a la estructura de los temas.

Por lo anteriormente expuesto se plantea el siguiente **problema a resolver**: ¿Cómo facilitar el proceso de gestión de temas de interfaz de usuario para el subsistema Bachiller de la Plataforma Educativa Zera?

A partir del problema se deriva como **objeto de estudio**: proceso de gestión de temas de interfaz de usuario para aplicaciones web.

Para dar solución al problema descrito se plantea como **objetivo general** de la investigación: Desarrollar un módulo que facilite el proceso de gestión de temas de interfaz de usuario para el subsistema Bachiller de la Plataforma Educativa Zera. Por consiguiente el **campo de acción** se reduce a la gestión de temas de interfaz de usuario para la Plataforma Educativa Zera.



Los **objetivos específicos** que permitirán dar cumplimiento al objetivo general son:

- ✓ Establecer los referentes teóricos de la investigación a partir del análisis de las herramientas de gestión de temas de interfaz de usuario.
- ✓ Realizar el análisis y diseño del módulo de gestión de temas de interfaz de usuario para el subsistema Bachiller.
- ✓ Implementar las funcionalidades del módulo teniendo en cuenta el análisis y diseño realizado.
- ✓ Realizar pruebas al módulo que permitan evaluar la calidad y funcionamiento del mismo.

Para dar cumplimiento a los objetivos planteados se determinan las siguientes **tareas de la investigación**:

- ✓ Realizar un estudio del estado del arte sobre las tendencias actuales y estándares de las herramientas de gestión de temas de interfaz de usuario.
- ✓ Analizar y definir lenguajes, herramientas y metodologías de desarrollo a utilizar.
- ✓ Identificar y especificar los casos de uso asociados a la gestión de temas de interfaz de usuario.
- ✓ Realizar el diagrama de casos de uso que especifica los requisitos funcionales y no funcionales.
- ✓ Realizar los diagramas de clases de análisis y diseño para cada caso de uso del sistema.
- ✓ Modelar diagrama de componente del sistema.
- ✓ Modelar diagrama de despliegue del sistema.
- ✓ Implementar las clases definidas en el modelo de clases del diseño.
- ✓ Diseñar los casos de prueba basado en los casos de uso.

A partir de los objetivos trazados se plantea la siguiente **idea a defender**: El desarrollo de un módulo para la gestión de temas de interfaz de usuario para el subsistema Bachiller facilitará este proceso en la Plataforma Educativa Zera.



A continuación se mencionan los métodos científicos utilizados para el desarrollo de la investigación.

Métodos Teóricos:

Histórico – lógico: es utilizado para realizar revisiones críticas sobre la bibliografía existente y conformar un criterio sólido sobre el estado de las investigaciones en el área de la gestión de temas de interfaz de usuario.

Analítico – sintético: es utilizado en el estudio de los conceptos relacionados con la gestión de temas de interfaz de usuario y los sistemas similares existentes, para luego sintetizar toda la información analizada y conformar la propuesta de solución.

Modelación: se emplea para realizar los casos de usos identificados que dan cumplimiento a los requisitos funcionales y no funcionales de la solución, así como la realización de modelos que brindan una visión general de la solución.

Como **resultado** se espera obtener un módulo con funcionalidades que faciliten el proceso de gestión de temas de interfaz de usuario desde la Plataforma Educativa Zera.

La presente investigación cuenta con cuatro capítulos que contienen la estructura de la investigación, el capítulo uno permite obtener la fundamentación teórica que soporta la investigación, sobre la gestión de temas de interfaz de usuario. En el capítulo dos se presentan las características del sistema y se obtienen los requisitos funcionales y no funcionales. En el capítulo tres se realiza el análisis y diseño de las funcionalidades establecidas en el capítulo anterior y en el capítulo cuatro se lleva a cabo la implementación y prueba de los requisitos obtenidos durante el desarrollo del sistema.



Capítulo I: Fundamentación teórica de la gestión de temas

Introducción

En este capítulo se especifica la fundamentación teórica que soporta la investigación sobre la gestión de temas de interfaz de usuario, la cual incluye el estudio del estado del arte, donde se realiza el análisis de soluciones existentes. Además se lleva a cabo el estudio de las tendencias en las tecnologías, metodologías y herramientas actuales, en las que se apoya la solución de la problemática a resolver.

1.1 Principales conceptos de la investigación

1.1.1 Interfaz

El diccionario de la Real Academia de la Lengua Española (5) define interfaz como una palabra derivada del término inglés —interface (superficie de contacto) y la define de la siguiente manera: “conexión física y funcional entre dos aparatos o sistemas independientes.”

El diccionario El Mundo (6) define: “una interfaz es el punto, el área, o la superficie a lo largo de la cual dos objetos de naturaleza distinta convergen.”

A partir de las definiciones analizadas anteriormente se concluye que la presentada por la Real Academia de la Lengua Española es la que se adecua a las características de la investigación.

1.1.2 Interfaz de usuario

A partir de la definición anterior, cuando uno de los sistemas que se comunica es un ser humano, se analiza entonces al concepto de interfaz de usuario. Por un lado se tiene un sistema físico o informático y por otro a una persona que desea interactuar con él, siendo la interfaz de usuario la herramienta que entiende a ambos y es capaz de traducir los mensajes que se intercambian.

El diccionario Glosarium (7) define: “Interfaz de Usuario es la parte de una aplicación que se encarga de interactuar con el usuario, una conexión e interacción entre hardware, software y usuario, es decir como la plataforma o medio de comunicación entre usuario o programa.”

Según Ben Shneiderman (8): “La interfaz de usuario es la parte de cualquier sistema que determina el control y la forma en que las personas operan sobre el equipo. Cuando la interfaz está bien diseñada, es



comprensible, previsible y controlable, los usuarios se sienten competentes, satisfechos y responsables de sus actos.”

A partir del análisis de las definiciones anteriores se concluye que el concepto guía para el desarrollo de la investigación es el de Shneiderman, pues establece los aspectos esenciales de una interfaz bien definida y las partes involucradas en su construcción.

La interfaz de usuario presenta una serie de clasificaciones, según su construcción pueden ser de software o de hardware (9):

- ✓ Interfaces de hardware: a nivel de los dispositivos utilizados para ingresar, procesar y entregar los datos: teclado, ratón y pantalla.
- ✓ Interfaces de software: destinada a entregar información acerca de los procesos y herramientas de control, a través de lo que el usuario observa habitualmente en la pantalla.

Atendiendo a como el usuario puede interactuar con una interfaz, se conocen varios tipos de interfaces de usuario (9):

- ✓ Interfaces de líneas de comando: es el estilo más antiguo de interacción hombre-máquina. El usuario escribe órdenes utilizando un lenguaje formal con un vocabulario y una sintaxis propia. Se utiliza un teclado, típicamente, y las órdenes están encaminadas a realizar una acción.
- ✓ Interfaces de menú: un menú es una lista de opciones que se muestran en la pantalla para que los usuarios elijan la opción que deseen. Las interfaces de menús aparecen cuando el ordenador se vuelve una herramienta de usuario y no sólo de programadores. Las actuales interfaces gráficas u orientadas a objetos siguen utilizando este tipo de interfaces.
- ✓ Interfaces gráficas de usuario: es una representación gráfica en la pantalla del ordenador, de los programas, datos y objetos, así como de la interacción con ellos. Proporciona al usuario las herramientas para realizar sus operaciones, más que una lista de las posibles operaciones que el ordenador es capaz de hacer.

Por las características en la presentación la información, la investigación basará su estudio en las interfaces gráficas de usuario.



1.1.3 Ventajas de una apropiada interfaz gráfica de usuario

El desarrollo de una apropiada interfaz debe permitir a los usuarios interactuar con el sistema independientemente de las características del individuo y de su situación geográfica, por lo que deben presentar tres aspectos esenciales en su desarrollo.

Usabilidad

El grado de usabilidad de un sistema interactivo es inversamente proporcional al tiempo que malgastan los usuarios de dicho sistema intentando averiguar el alcance de qué hace o dónde se encuentra una determinada funcionalidad. Esta característica hace, por tanto, referencia a la rapidez y facilidad con que las personas llevan a cabo sus tareas a través del uso del producto con el que están trabajando.

El concepto de usabilidad puede ser definido, además como atributo de calidad de una aplicación, consecuentemente, como disciplina o enfoque de diseño y evaluación. A partir de estas conceptualizaciones, se obtienen los principios básicos en los que se basa la usabilidad conocidos como atributos de usabilidad o características orientadas al usuario (10):

- ✓ **Facilidad de aprendizaje:** cuán fácil es aprender la funcionalidad básica del sistema y ser capaz de realizar correctamente la tarea que desea realizar el usuario.
- ✓ **Eficiencia:** el número de transacciones por unidad de tiempo que el usuario puede realizar usando el sistema. Este atributo busca la máxima velocidad de realización de tareas del usuario. Cuanto mayor es la usabilidad de un sistema, más rápido es el usuario al utilizarlo y el trabajo se realiza con mayor rapidez.
- ✓ **Recuerdo en el tiempo:** refleja el recuerdo acerca de cómo funciona el sistema que mantiene el usuario, cuando vuelve a utilizarlo tras un período de no utilización.
- ✓ **Tasa de errores:** este atributo contribuye de forma negativa a la usabilidad de un sistema. Se refiere al número de errores cometidos por el usuario mientras realiza una determinada tarea. Un buen nivel de usabilidad implica una tasa de errores baja. Los errores reducen la eficiencia y satisfacción del usuario y pueden verse como un fracaso en la transmisión al usuario del modo de interactuar con el sistema.
- ✓ **Satisfacción:** éste es el atributo más subjetivo. Muestra la impresión subjetiva que el usuario obtiene del sistema.



Accesibilidad

Un concepto íntimamente ligado al de usabilidad es el de accesibilidad. Este ya no se refiere a la facilidad de uso, sino a la posibilidad de acceso. En concreto a que el diseño, como prerequisite imprescindible para ser usable, posibilite el acceso a todos sus usuarios, sin excluir a aquellos con limitaciones individuales, discapacidades, dominio del idioma o limitaciones derivadas del contexto de acceso.

El concepto se utiliza para nombrar al grado en que las personas, más allá de sus capacidades físicas o técnicas, pueden utilizar un cierto objeto o acceder a un servicio. Los seres humanos son diferentes entre sí y las interfaces de usuario deben acomodarse a esas diferencias, de tal modo que cualquier persona sea capaz de utilizarlas sin que existan limitaciones. Es necesario evitar diseñar solamente atendiendo a características de grupos de población específicos, imponiendo barreras innecesarias que podrían ser evitadas prestando más atención a las limitaciones de estos (11).

Personalización

El término personalización intenta acercar los intereses y las necesidades del usuario, la demanda, a la oferta de productos y servicios. La personalización es como el resultado de establecer un diálogo con el cliente para ofrecer un mejor servicio. El objetivo de la personalización consiste en responder personalizadamente a las necesidades del usuario. Dar un mejor servicio al cliente ocurre al mostrar un servicio interactivo, al establecer un diálogo con el cliente.

El establecimiento de un diálogo con el cliente permitirá obtener un mayor conocimiento de sus necesidades, donde éste se convierte en cocreador de la oferta. La participación del cliente en la elaboración del producto significa una personalización del mismo (12).

La interfaz de usuario independientemente de la finalidad con que es concebida debe tener en cuenta estas características, lo que garantizará facilidad al usuario, además de una adecuada presentación de la información.

1.1.4 Temas de interfaz de usuario

En el sitio *Ámbito web* (13) definen un tema como: “un frontal intercambiable de un nodo o grupo de nodos, que controla elementos tales como el mensaje de cabecera, la navegación y el aspecto de un nodo. La opción de establecer un tema sólo está disponible al crear un nodo raíz y los nodos anidados inmediatamente por debajo de ese nodo raíz.”



Según Cristóbal González (14) experto de Drupal, un tema es: “la capa de presentación de la interfaz, es una colección de imágenes, CSS y archivos HTML/PHP que modifican la apariencia por defecto de cómo se muestra la información o contenido.”

A modo de resumen un tema es la representación de la interfaz de usuario, que contiene elementos que lo conforman como estilos, imágenes y archivos de configuración. El tema de interfaz de usuario es el aspecto visual del sistema.

1.1.5 Diseño de interfaz de usuario en plataformas educativas

Las plataformas son Entornos Virtuales de Enseñanza y Aprendizaje (EVEA), que surgen y se desarrollan con la intención de dar soporte a la docencia, se apoyan en sistemas informáticos basados habitualmente en protocolos de la web con herramientas adaptadas a las necesidades educacionales requeridas (15).

Tanto en la educación a distancia, como en la modalidad presencial, las plataformas educativas pueden contribuir a mejorar el aprendizaje. Se debe tener presente que el aprendizaje ocurre en la mente del individuo y no en los circuitos de un ordenador. El papel de la educación, junto con la ciencia y la tecnología es fundamental pero debe traer consigo un buen diseño de interfaz de usuario.

Según Herrera y Latapié (16), existen algunas funciones que competen el diseño en los nuevos entornos de aprendizaje:

- ✓ Manejo del lenguaje gráfico, propiedades y disposición en el espacio: se deben tener en cuenta los contraste de colores a utilizar, el equilibrio, la simetría, textura, tamaño y su ubicación en la interfaz. Estos elementos generan sensaciones específicas en el usuario (equilibrio-tensión o reposo-estrés) que pueden favorecer o inhibir el proceso de aprendizaje. Se debe evitar la ambigüedad en la disposición de los elementos visuales.
- ✓ Procesos cognitivos de aprendizaje: estos son activados a partir de los estímulos sensoriales que la interfaz provee al usuario, la información debe ser presentada de forma explícita y clara, sin redundancias, solo lo que el estudiante debe conservar para el apoyo a su aprendizaje.
- ✓ El bienestar y comodidad visual: no debe contener elementos perturbadores que distraigan o fatiguen al usuario innecesariamente. Las condiciones deben estar orientadas a facilitar sus actividades y hacer más cómoda y placentera su estancia virtual, por lo que toda información no deseable para el usuario debe ser evitada. Los factores que pueden incidir en esta característica



son la intensidad luminosa, contraste y color.

El estudio de las características que debe presentar la interfaz de usuario en plataformas educativas, sirve de apoyo para conocer los principales elementos a tratar dentro de la interfaz y los beneficios que presenta una adecuada presentación de los conocimientos a los estudiantes.

1.1.6 Estudio de sistemas similares que realizan la gestión de temas

La gestión de interfaz de usuario en la actualidad se ha ampliado, con la necesidad de facilitar el trabajo en los distintos entornos que proveen interacción directa con los usuarios. Estos entornos como los Sistemas de Gestión de Contenido (CMS, por sus siglas en inglés) y los Sistemas de Gestión del Aprendizaje (LMS, por sus siglas en inglés), proporcionan las herramientas necesarias para diseñar interfaces de usuario teniendo en cuenta las capacidades cognitivas y de percepción de las personas. Las más utilizadas a nivel mundial se presentan a continuación.

Drupal

Drupal es un CMS modular y muy configurable. Es un programa de código abierto, con licencia pública de GNU (GPL, por sus siglas en inglés) escrito en Procesador de Hipertexto (PHP, por sus siglas en inglés), desarrollado y mantenido por una activa comunidad de usuarios. Se destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web y un énfasis especial en la usabilidad y consistencia de todo el sistema (17).

Drupal permite configurar temas ya existentes a partir de la creación de subtemas, creando uno que utilice las funcionalidades y los estilos de un tema ya existente, sin tener que desarrollarlas nuevamente desde cero, todos los cambios que se añaden al subtema estarán por encima del tema base, ya que permite usar los elementos del tema base sin limitar las habilidades para configurar y modificar el nuevo subtema. (18)

Características de los temas en Drupal (19):

- ✓ Las plantillas en Drupal son matrices que definen como estará estructurado el contenido del sitio y con la modificación de un único archivo permite dar estilo a varias páginas.
- ✓ El archivo de información se encarga de definir las regiones del tema, que no es más que la forma en que Drupal define donde se podrá colocar el contenido.
- ✓ La construcción de un tema consiste en la definición de las reglas de Hojas de estilo en cascada (CSS, por sus siglas en inglés) para darle un aspecto visual al contenido.



Cada tema, además de los elementos de estilo visual, afecta:

- ✓ La presentación del menú principal.
- ✓ Las zonas posibles para ubicación del contenido.
- ✓ La ubicación de la caja de búsqueda.

Un tema es sencillo de personalizar, accediendo a los ficheros donde se define la apariencia y organización general de las páginas, las zonas, los bloques y la visualización de cada tipo de contenido.

Los bloques tienen un doble sistema de control de visualización, por lo que se puede hacer que el sitio web tenga cierta fluidez y controlar la sobrecarga de contenido. Un bloque puede mostrarse sólo en una determinada situación, definiendo positivo o negativo de modo general (20).

Desventajas:

La creación de un tema desde cero presenta una complejidad elevada, ya que posee mecanismos y reglas que son difíciles de comprender en su totalidad.

- ✓ Alta complejidad en el aprendizaje tanto para el uso como para el desarrollo.
- ✓ Contiene varios niveles interiores (submenús o sub categorías) en el panel, lo que puede traer consigo confusión en los desarrolladores.

WordPress

WordPress es un CMS con avanzada plataforma semántica de publicación personal, orientada a la estética, los estándares web y la usabilidad. WordPress es libre y al mismo tiempo, gratuito. Soporta plugins¹ y creación de páginas estáticas a través de la base de datos. Separa el contenido y el diseño en Lenguaje de Marcado de Hipertexto Extensible (XHTML, por sus siglas en inglés) y CSS lo que permite modificar de una manera sencilla el estilo del sitio (21).

WordPress es muy flexible, trabaja con plantillas para permitir mayor control de la interfaz del sitio, permite realizar varios cambios a estas plantillas desde el propio CMS. Para realizar modificaciones desde el panel de administración de WordPress es necesario contar con los permisos necesarios.

Características de un tema en WordPress (22):

¹ Es una aplicación que se relaciona con otra para aportarle una función nueva.



- ✓ Contiene un archivo central de todos los temas, desde el cual se accede a los otros archivos del tema y contiene el medio para obtener la información desde la base de datos.
- ✓ Contiene un archivo que es el encargado de abrir la página e incluye además el titular y el menú horizontal de la página, si tuviera.
- ✓ Contiene un archivo para el pie de la web, los créditos y dónde opcionalmente se puede incluir la función que permite realizar tareas o imprimir código en Lenguaje de Marcado de Hipertexto (HTML, por sus siglas en inglés) al final de la página, es muy utilizada para modificar o agregar código una vez que toda la página está cargada.
- ✓ Contiene la hoja de estilo principal, es obligatoria, además de definir la información de estilos para la web, es dónde se ubica la información sobre el tema.

Desventajas:

- ✓ Para modificar la estructura y forma en que se muestra un tema, se debe tener conocimiento sobre el lenguaje de desarrollo.
- ✓ Al crear un subtema se realiza una llamada a otro tema, esto puede afectar el rendimiento de la web.
- ✓ No permite modificar el código fácilmente, se pueden agregar Widgets² ya existentes, pero no permite modificarlos completamente.

Joomla

Joomla es un CMS que permite construir sitios web y aplicaciones en línea de gran alcance. Muchos aspectos, incluyendo su facilidad de uso y extensibilidad, han hecho de Joomla el software disponible, más popular para la construcción de sitios web. Joomla es una solución de código abierto que está disponible gratuitamente para todo el mundo. Está diseñado para ser fácil de instalar y configurar, incluso si no es un usuario avanzado, es altamente extensible y posee varias extensiones de forma gratuita bajo la GPL, está disponible en el directorio de extensiones de Joomla (23).

Características de Joomla:

- ✓ Las secciones de temas pueden ampliarse mediante aportaciones de autores.
- ✓ Las características de etiquetado de los contenidos permiten un acceso flexible para cada tipo de

² Pequeña aplicación o programa que provee de información visual a los usuarios.



usuario.

- ✓ Se puede modificar completamente el sitio y la administración con las plantillas simples de página única.
- ✓ Los diseños pueden ser completamente personalizables, incluyendo los menús izquierdo, derecho y central.
- ✓ Contiene una plantilla totalmente accesible en apoyo de las personas con distintos tipos de discapacidad.
- ✓ Con los estilos de plantilla se pueden crear fácilmente pequeñas variaciones de plantilla, que puede ser asignadas a páginas específicas en el sitio web.
- ✓ Administración sencilla.

Desventajas:

- ✓ No presenta comentarios por defecto.
- ✓ Es difícil declarar una plantilla para algunos casos como en WordPress.
- ✓ No todas las extensiones son compatibles con las nuevas versiones y generan conflicto con algunas otras extensiones.
- ✓ Requiere programación más compleja para mayores resultados satisfactorios.
- ✓ Se puede crear una sola plantilla para todas las páginas, a menos que se declaren más argumentos.

MOODLE

El Entorno Modular de Aprendizaje Dinámico Orientado a Objetos (MOODLE, por sus siglas en inglés), es un entorno de formación denominado LMS o plataforma de teleformación, con licencia GPL, por lo que se puede calificar como software libre. El principal rasgo característico de este tipo de entorno o aplicación es que permite gestionar y desarrollar un proceso de enseñanza-aprendizaje a través de internet, utilizando para ello una serie de herramientas de comunicación, trabajo colaborativo, gestión y seguimiento de los alumnos de forma integrada (24).

MOODLE tiene un potente sistema de temas que permite una variedad de efectos a través del uso de XHTML y CSS. Los temas se pueden basar en el tema estándar, que es muy simple pero funcional. Se pueden modificar los estilos simplemente agregando hojas de estilo en el propio tema, si aumentan más



adelante y los nuevos estilos son necesarios, el tema seguirá trabajando sin ningún cambio, pues las nuevas clases serán definidas en el tema estándar.

Características de los temas en MOODLE (25):

- ✓ Los temas pueden ser configurados a nivel de sitio, curso y usuario.
- ✓ Cada página se maneja de forma individual por CSS, permitiendo apuntar exactamente a artículos.
- ✓ Los temas se pueden basar en el tema estándar, que es muy simple pero funcional.
- ✓ Los temas se pueden basar además en cualquier otro tema, permitiendo que se cree fácilmente familias de temas, o las variaciones de un tema.
- ✓ Permite crear una familia de temas diferentes, coloreados para los propósitos de la accesibilidad.
- ✓ Para crear un tema propio es preferible crear uno nuevo, con su propia carpeta, y basar el tema en uno existente. Si sólo se desea modificar un tema de los disponibles, los cambios que se hagan se perderán en la próxima actualización.

Desventajas:

- ✓ Al adaptarse a diferentes modelos de enseñanza se vuelve complejo de gestionar o parametrizar sin una previa preparación sobre su funcionamiento.
- ✓ La personalización adaptada a una imagen corporativa de un centro educativo se convierte en algo complejo que requiere de los servicios externos de personal especializado.

Luego de realizar el estudio de los sistemas similares, se observa que estas herramientas presentan una serie de criterios que son comunes para muchas de ellas. Su análisis servirá para la construcción del diseño de la propuesta de solución tomando de cada uno sus principales ventajas.

- ✓ Drupal brinda la opción que permite seleccionar un tema existente y modificarlo según las preferencias de los usuarios y la página de configuración individual para cada tema.
- ✓ WordPress brinda la opción que permite que las actualizaciones del tema principal no afectan a los estilos y modificaciones de un subtema.
- ✓ Joomla tiene como ventaja que los diseños pueden ser completamente personalizables, incluyendo los menús.
- ✓ MOODLE brinda la opción de que los temas se basan en el tema estándar, que es muy simple pero funcional.



Estas herramientas sirvieron de apoyo pero ninguna puede ser utilizada en la solución pues no cumplen con los requisitos de implementación de la plataforma, además que no cuentan con la funcionalidad que permite posicionar los Widgets dentro de cada tema, siendo esta característica esencial en la gestión de temas de interfaz de usuario para Bachiller.

1.2 Ambiente de desarrollo

Al iniciar el desarrollo de la Plataforma Educativa Zera se llevó a cabo un estudio de las tecnologías, lenguajes y herramientas de modelado, servidores web, frameworks y lenguajes de desarrollo con el objetivo de definir el ambiente de desarrollo de la misma. Para obtener un sistema con la calidad requerida se decide trabajar sobre el ambiente de desarrollo ya definido. A continuación se detallarán las características del mismo.

1.2.1 Metodología de desarrollo de software

Las metodologías o procesos de desarrollo de software ofrecen un compendio de procedimientos, reglas, herramientas, documentación y técnicas que controlan el proceso de desarrollo de un sistema informático. Permite la creación de aplicaciones flexibles y robustas de un modo organizado, facilitando su comprensión. Cada proyecto tiene características propias, por ello existen varias metodologías, lo cual permite escoger la que más se adapte a las necesidades y recursos de cada equipo de desarrollo.

Existen dos tipos de metodologías: ágiles y tradicionales, esta última se caracteriza por ser (...) “una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente” (26), siendo esta la utilizada en la Plataforma Educativa Zera.

Proceso Unificado de Software

El Proceso Unificado de Software (RUP, por sus siglas en inglés) es una metodología robusta que provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte (27).

El ciclo de vida de RUP se caracteriza por:

- ✓ Guiado por casos de uso: los casos de uso son el instrumento para la descripción funcional de lo



requerido por los usuarios finales y constituyen la guía fundamental para las actividades a realizar durante todo el proceso de desarrollo del sistema.

- ✓ Centrado en la arquitectura: la arquitectura muestra una visión completa del sistema, para que todos los implicados en el desarrollo tengan una idea clara de qué es lo que están realizando.
- ✓ Iterativo e incremental: durante todo el proceso de desarrollo se producen versiones superiores.

Define cuatro fases esenciales (Inicio, Elaboración, Construcción y Transición) y nueve flujos de trabajos; seis de Ingeniería (Modelado del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue) y tres de apoyo (Configuración y manejo de cambios, Entorno y Gerenciamiento de proyecto) (28).

Con la utilización de RUP se garantiza un modelado visual del sistema obteniendo la documentación necesaria de los artefactos que genera, sirviendo para el futuro desarrollo de la solución.

1.2.2 Lenguaje de modelado

Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es utilizado para especificar, visualizar, construir y documentar los artefactos de un sistema de software (28).

Los principales beneficios de UML son:

- ✓ Mejores tiempos totales de desarrollo.
- ✓ Modelar sistemas utilizando conceptos orientados a objetos.
- ✓ Establecer conceptos y artefactos ejecutables.
- ✓ Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- ✓ Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- ✓ Mejor soporte a la planeación y al control de proyectos.
- ✓ Alta reutilización y minimización de costos.

UML divide cada proyecto en un número de diagramas que representan las distintas vistas del proyecto y juntos representan la arquitectura del mismo. Permite describir un sistema en diferentes niveles de abstracción, simplificando la complejidad sin perder información, para que los usuarios y desarrolladores comprendan las características de la aplicación.



Mediante UML se podrán especificar las decisiones tomadas en el análisis, diseño e implementación de la solución, construyéndose de esta forma modelos completos y permitirá documentar todos los artefactos del proceso de desarrollo. Se seleccionó la versión UML 2.1 como lenguaje de modelado.

1.2.3 Lenguajes de desarrollo

Lenguajes del lado del cliente

❖ XHTML

XHTML es una familia de tipos de documentos que reproducen y extienden el lenguaje HTML 4. Ante la llegada al mercado de un gran número de dispositivos, XHTML surge como el lenguaje cuyo etiquetado, más estricto que HTML, va a permitir una correcta interpretación de la información independientemente del dispositivo desde el que se accede a ella. XHTML, al estar orientado al uso de un etiquetado correcto, exige una serie de requisitos básicos a cumplir dentro del código. Entre estos requisitos básicos se puede mencionar, una estructuración coherente dentro del documento donde se incluirían elementos correctamente anidados, etiquetas en minúsculas, elementos cerrados correctamente y atributos de valores entrecomillados (29).

Es compatible con los navegadores HTML, aportando de forma sencilla todas las ventajas del Lenguaje de Marcado Extensible (XML, por sus siglas en inglés) a documentos HTML preexistentes. Puede ser integrado con otras tecnologías XML, los documentos pueden visualizarse en navegadores HTML y utilizar el Modelo de Objeto del Documento (DOM, por sus siglas en inglés) de HTML o el de XML. Puede extenderse o reducirse hasta acomodarse a las necesidades de la aplicación (30). Se utilizará XHTML en su versión 1.0 para el desarrollo del sistema.

❖ CSS

CSS es un mecanismo simple que describe cómo se muestra un documento en la pantalla, cómo se imprime y cómo se pronuncia la información a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilos y formatos de sus documentos. Los CSS son utilizados para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML y XML. Permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento CSS afectará a todas las páginas vinculadas a él en las



que aparezca el elemento al que hace referencia (31). Se utilizará CSS en su versión 3.0 para los estilos del sistema.

❖ **Java Script**

Java Script es un lenguaje de programación interpretado por el navegador, desarrollado para incrementar las funcionalidades de HTML y XHTML. Permite incluir macros en páginas web, estas se ejecutan en el ordenador del visitante de las páginas y no en el servidor. Se utiliza para controlar las ventanas del navegador y el contenido que muestran, evitar depender del servidor web para cálculos sencillos, capturar los eventos generados por el usuario y responder a ellos sin salir a Internet, y comprobar los datos que el usuario introduce en un formulario antes de ser enviados (32).

Tecnología del lado del cliente

❖ **AJAX**

Java Script Asíncrono y XML (AJAX, por sus siglas en inglés), es un conjunto de tecnologías aplicadas de forma concreta, permite crear aplicaciones web más eficientes en la interacción con el usuario.

Asocia las tecnologías (33):

- ✓ Estructura y presentación de la información basada en estándares XHTML y CSS.
- ✓ DOM para interactuar dinámicamente con los datos.
- ✓ Intercambio y manipulación de datos usando XML.
- ✓ Recuperación de datos asíncronos utilizando XML y Java Script.

Dentro de sus principales ventajas se encuentra:

- ✓ Rapidez en las operaciones.
- ✓ Más cerca de crear realmente aplicaciones web.
- ✓ Menos carga del servidor (menos transferencia).
- ✓ Menos ancho de banda.

Esta tecnología será utilizada en el desarrollo para la comunicación asíncrona³ de datos.

³ Asíncrona: los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.



Lenguajes del lado del servidor

❖ PHP

Entre las ventajas más destacadas de PHP se puede mencionar que es un lenguaje completamente libre por lo que se destaca por tener una amplia comunidad, dándole un mayor soporte y mejorando el tiempo de recuperación ante errores. La documentación del lenguaje puede ser encontrada en varios idiomas por lo que mejora la agilidad en el desarrollo.

Facilita la realización en pocas líneas de código y trabaja en combinación con otras tecnologías como Perl, Java Script y Python. Es muy legible y fácil de aprender, no soporta directamente punteros, de forma que no existen los problemas de depuración provocados por estos y es de fácil acceso para todos (34). Este lenguaje se utilizará para interactuar con los datos de la aplicación y se decide utilizar su versión 5.3.10.

❖ XML

Es un lenguaje muy simple pero estricto, que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML permite la lectura de datos a través de diferentes aplicaciones que lo utilicen (35).

Este lenguaje será utilizado para almacenar toda la información referente al estándar a implementar. Se empleará su versión 1.0.

1.2.4 Frameworks

Un framework web puede ser definido como un conjunto de componentes que disponen de un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar el código ya existente y promover buenas prácticas de desarrollo con el uso de patrones.



Capa de Presentación

❖ Framework Java Script

JQuery es un framework de Java Script, que permite simplificar la interacción con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.

Dentro de sus ventajas se encuentra el soporte a extensiones o plugins los cuales amplían las facilidades de uso y mejoran los tiempos de producción al poder reutilizar componentes desarrollados por terceras personas (36).

Se selecciona como parte del desarrollo de la investigación JQuery en su versión 1.9.

❖ Framework CSS

Un framework CSS está diseñado para apoyar el desarrollo de software. Este tipo de framework intenta aliviar el exceso de carga asociado con actividades comunes al trabajar con hojas de estilos y su intención es principalmente ahorrar tiempo y minimizar el riesgo de errores en el desarrollo de aplicaciones. Los frameworks evitan, sobre todo en los momentos iniciales del desarrollo de una aplicación, repetir código para construir la base, permiten conseguir ciertas funcionalidades y se componen de uno o varios archivos con declaraciones predefinidas (37).

Ventajas de usar framework CSS:

- ✓ Permite agilizar el desarrollo, especialmente en sus momentos iniciales.
- ✓ Se comienza a partir de una base nivelada sobre la que desarrollar un trabajo adicional.
- ✓ Si el framework CSS está bien documentado, agiliza el trabajo en un equipo relativamente grande.

Existe gran variedad de frameworks CSS, a continuación se expondrán los más utilizados y las características principales de cada uno:

Blueprint

Es un framework CSS que pretende reducir el tiempo de desarrollo de las páginas web. Ofrece una estructura sólida en la que funda el trabajo de diseño y maquetación, por medio de las rejillas. Ofrece una serie de clases muy útiles para estilizar componentes típicos como formularios, botones, pestañas, tipografías o para páginas web que puedan imprimir de manera óptima (38).

Características de Blueprint (39):



- ✓ Una rejilla, que permitirá crear cualquier estructura de página web. Básicamente se trata de un conjunto de clases CSS para posicionar cualquier elemento en un espacio de 950 píxeles de anchura, dividido en 24 columnas.
- ✓ Una definición de tipografía predeterminada, de manera precisa para todos los elementos de la página que puedan tener textos.
- ✓ Estilos para formularios, con los que mejorar las interfaces de usuario.
- ✓ Estilos para impresión, con las definiciones CSS más útiles para que las páginas puedan ser leídas cuando se imprimen.
- ✓ Estilos específicos para Internet Explorer, con los que resolver algunos de los problemas más comunes.

Bootstrap

Es un framework desarrollado y liberado por Twitter. Utiliza componentes y servicios creados por la comunidad como HTML5, LESS, JQuery UI entre otras. Esta herramienta es un compendio de buenas prácticas, implementa los nuevos estándares HTML 5 y CSS 3. Es ágil para la construcción de interfaces, y para prototipar aplicaciones web en ciertos casos.

En principio fue creado solo para los navegadores modernos, Bootstrap ha evolucionado para incluir soporte para un mayor número de navegadores, para tabletas y telefonía móvil. Los componentes se adaptan y escalan según las resoluciones de pantalla y dispositivos para proporcionar una experiencia consistente (40).

Características de Bootstrap:

- ✓ Contiene una columna de respuesta de 12 rejillas.
- ✓ Contiene docenas de componentes.
- ✓ Utiliza plugins Java Script, tipografías y controles de formulario.
- ✓ Contiene un personalizador basado en la web para hacer su propia rutina de carga.

Al analizar las características de cada uno de los frameworks CSS se escoge Bootstrap en su versión 2.2.2 para ser utilizado en el desarrollo de la aplicación, pues presenta características esenciales que ayudarán a la personalización más efectiva de los temas, es adaptable a las resoluciones de pantalla, es ágil en la construcción de interfaces y contiene una gran cantidad de componentes.



Capa de Lógica de Negocio

Symfony es un framework basado en lenguaje PHP, que facilita el desarrollo de aplicaciones web. Se encarga de los aspectos comunes, dejando al programador aportar valor al desarrollar las características específicas de cada proyecto. Aumenta exponencialmente la productividad y ayuda a mejorar la calidad de las aplicaciones, implementando las buenas prácticas y patrones de diseño que se han definido para la web.

Características del framework (41):

- ✓ Fácil de instalar y configurar en sistemas Windows, Mac y Linux.
- ✓ Funciona con todas las bases de datos comunes (MySQL, PostgreSQL, SQLite, Oracle, MS SQL Server).
- ✓ Compatible solamente con PHP 5, para asegurar el mayor rendimiento y acceso a las características más avanzadas de PHP.
- ✓ Basado en la premisa de convenir en vez de configurar, en la que el desarrollador solo debe configurar aquello que no es convencional.
- ✓ Preparado para aplicaciones empresariales, puede adaptarse con facilidad a las políticas y arquitecturas propias de cada empresa u organización.
- ✓ Flexible y extensible mediante un completo mecanismo de plugins.
- ✓ Publicado bajo licencia de software libre y apoyado por una empresa comprometida con su desarrollo.
- ✓ Traducido a más de 40 idiomas y fácilmente traducible a cualquier otro.

El empleo de Symfony facilita la retroalimentación, superación y reparación de errores debido a que la comunidad de usuarios y documentación es muy amplia, los usuarios proveen diversos plugins que pueden ser reutilizables y esa es una de sus grandes fortalezas. Se decide utilizar su versión 1.4.2.

1.2.5 Servidor web

El servidor web Apache constituye uno de los más utilizados entre los servidores web por su estabilidad, confiabilidad y el gran aporte del grupo de voluntarios que planean y desarrollan todo lo relativo a esta plataforma, desde la documentación hasta el código.

Entre las ventajas que presenta Apache se encuentran (42):



- ✓ Personalizable: la arquitectura modular de Apache permite construir un servidor especializado, permitiendo la implementación de los últimos y más novedosos protocolos.
- ✓ En cuanto a la administración, los archivos de configuración de Apache están en ASCII, por lo que tiene un formato simple y pueden ser editados utilizando solamente un editor de texto. Estos archivos son transferibles permitiendo la clonación efectiva de un servidor.
- ✓ Puede ser administrado a través de línea de comandos, lo que hace la administración remota muy conveniente.
- ✓ Se trata de un servidor muy eficiente, consume menos recursos de sistema en comparación a otros servidores. Además puede ser ejecutado en varios sistemas operativos, dentro de ellos, varias versiones de UNIX, Windows9x/NT y MacOS.
- ✓ El soporte de Apache es provisto por la fundación Apache, constituida por usuarios dedicados a su comunidad, así como compañías que ofrecen versiones pagadas.

Apache será el servidor web a utilizar en la presente investigación en su versión 2.2.22. Se caracteriza por ser de código abierto, multiplataforma, soportar PHP y contar con una alta comunidad de usuarios lo que posibilita que se dé solución a muchos errores que se puedan presentar durante el proceso de desarrollo.

1.2.6 Entorno de desarrollo

Los Entornos de Desarrollo Integrado (IDE, por sus siglas en inglés) son programas con un conjunto de herramientas que brindan amplias comodidades en el desarrollo de aplicaciones. Las ventajas de usar IDE en el desarrollo de la problemática planteada son: completamiento de código, disminución en el tiempo de desarrollo, depuración de código y resaltado de sintaxis.

NetBeans

NetBeans es un IDE de código abierto, el cual soporta el desarrollo de todos los tipos de aplicaciones Java. Existe un número importante de módulos para comprenderlo, es un producto libre y gratuito sin restricciones de uso. Entre sus características se encuentra un sistema de proyectos basado en control de versiones y refactorización. Todas las funciones del IDE son provistas por plugins (43).

Sus características principales son (44):

- ✓ Compatibilidad total con PHP 5.3: espacios de nombre, funciones y clausuras, condiciones



ternarias y etiquetas de salto.

- ✓ Compatibilidad con el framework Symfony: proyectos, comandos de Symfony y métodos.
- ✓ Abreviados de teclado y resaltado de sintaxis PHP en archivos YAML.
- ✓ Creación de proyectos PHP desde aplicaciones PHP remotas.
- ✓ Mejoras en la integración de Protocolo de Transferencia de Archivo (FTP, por sus siglas en inglés) y exclusión de las carpetas del proyecto PHP de exploraciones e indexaciones.

NetBeans brinda soporte de forma nativa a frameworks como jQuery y Symfony, anteriormente seleccionado para el desarrollo del sistema, por lo se escoge como IDE a utilizar en su versión 7.1.

1.2.7 Herramientas para el modelado del sistema

La Ingeniería de Software Asistida por Computadora (CASE, por sus siglas en inglés) apoya el avance de los aspectos del ciclo de vida de desarrollo del software, en tareas como la realización del diseño del proyectos, cálculo de costes, implementación de una porción del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras. Las herramientas CASE de modelado con UML permiten analizar y diseñar orientado a objetos y abstraer el código fuente, a un nivel donde la arquitectura y el diseño se tornan más obvios y más fáciles de entender y modificar.

Visual Paradigm para UML

Visual Paradigm para UML es una herramienta CASE creada para el ciclo vital del desarrollo de software, permitiendo la captura de requisitos, análisis, diseño e implementación. Considerada como una herramienta muy completa y fácil de usar, con soporte multiplataforma. Es compatible con todas las fases y roles en un proceso de desarrollo de software tales como el ingeniero de software, analistas del sistema, arquitecto del sistema, entre otros. Visual Paradigm para UML es empleado para las aplicaciones grandes y complejas que presentan un enfoque orientado a objeto, proporciona características tales como generación de código, ingeniería inversa así como generación de bases de datos y diagramas de flujo de datos. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases (45).

Se ha seleccionado Visual Paradigm 8.0 como herramienta CASE para el modelado del sistema.



1.2.8 Herramienta de prototipado

En el mundo del diseño web las herramientas de prototipado son la representación esquemática de una página web, sin elementos gráficos que muestran contenido y comportamiento de las páginas. Son realmente útiles entre programadores, diseñadores y clientes. Son sumamente importantes para crear los bocetos del sistema a desarrollar, centrando de esta forma la atención en la arquitectura de la información.

Balsamiq Mockups

Balsamiq es una herramienta para diseñar prototipos para el cliente, que permite proyectar los requerimientos del usuario de tal forma que se puedan presentar bocetos que se acerquen a la realidad, contemplando explicaciones o aclaraciones sobre las distintas funcionalidades que se pueden dar en cada caso. Es una herramienta muy sencilla de usar a la hora de crear representaciones esquemáticas.

Cuenta con una colección de más de 75 modelos a utilizar, contiene desde aplicaciones para móviles hasta cualquier aplicación web. Dispone además de una colección muy amplia de contribuciones de usuarios en las que se pueden encontrar una gran variedad de prototipos. Para el cliente es muy clarificador y los bocetos pueden ser utilizados como base para extraer los requisitos del usuario y realizar el análisis (46).

Se selecciona como herramienta de prototipado Balsamiq para la realización de los bocetos, pues su principal ventaja es que es rápida y muy fácil de utilizar, permitiendo crear los bocetos en muy poco tiempo.

1.2.9 Analizador de archivos CSS

Sabberworm

Es un analizador de archivos CSS que permite la extracción en una estructura de datos, la manipulación de dicha estructura y de salida como CSS optimizado. Al utilizar el analizador se crea una nueva instancia y la estructura de datos resultantes consiste principalmente en listas CSS, contenedor genérico que presenta bloques de declaraciones (conjunto de reglas con selectores) (47).

El analizador de archivos CSS Sabberworm es el escogido para permitir la manipulación de los estilos en el sistema, ya que garantiza una interacción sencilla y completa con los archivos CSS.



Conclusiones del capítulo

En el capítulo se realizó un resumen de los principales conceptos asociados al dominio del problema, con el objetivo de lograr un amplio conocimiento del mismo y explicar de forma argumentada el objeto de estudio de la investigación. Se realizó un análisis de sistemas similares, sirviendo de apoyo al desarrollo de la aplicación. Además se llevó a cabo un análisis de las tecnologías, lenguajes y herramientas que servirán para el desarrollo de la investigación.



Capítulo II: Características del sistema

Introducción

En este capítulo se presentan las características del sistema llegando a un acuerdo entre clientes y otros interesados, sobre lo que el sistema debe hacer para la elaboración del Modelo de Dominio. Provee a los desarrolladores un mejor entendimiento de los requisitos del sistema, definiéndose los actores y las relaciones entre ellos, así como los casos de uso con sus descripciones textuales.

2.1 Modelo de dominio

Una de las actividades centrales del ciclo de desarrollo consiste en crear un modelo conceptual para los casos de uso, donde se explican los conceptos significativos en un dominio del problema.

El Modelo de dominio también conocido como Modelo conceptual es la representación visual de los conceptos u objetos del mundo real en un dominio de interés. Este modelo agrupa los conceptos de un dominio, es el mecanismo fundamental para comprender el problema y para establecer conceptos comunes (48). Se representa en UML con un diagrama de clases en el que se muestra: conceptos u objetos del dominio del problema, clases conceptuales y asociaciones entre las clases conceptuales. A continuación se muestra el Modelo de dominio de la presente investigación (Ver Figura 1. Modelo de dominio)

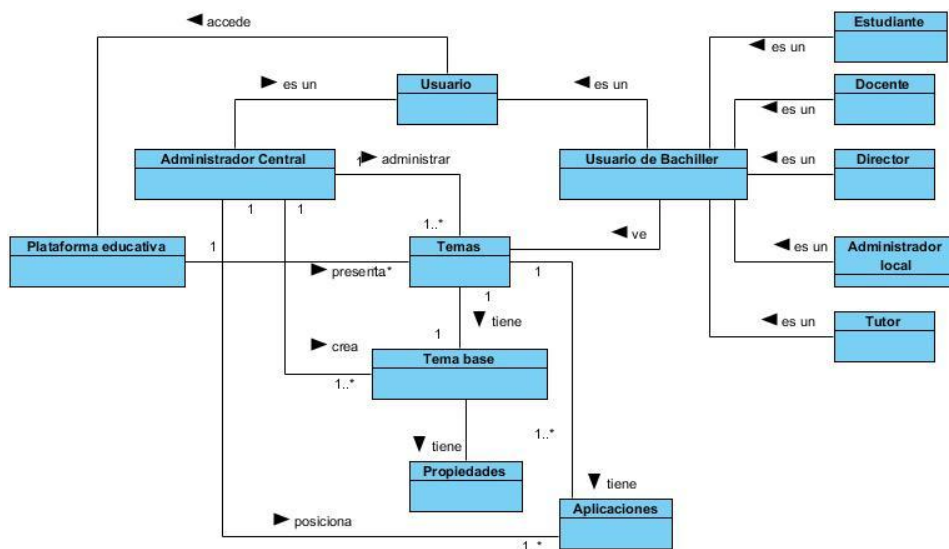


Figura 1. Modelo de dominio.

2.1.1 Descripción de los conceptos del dominio

Los conceptos relacionados con el negocio, especificados en el Modelo de Dominio ayudarán a los usuarios, clientes y desarrolladores a emplear un vocabulario común para entender el contexto en que se sitúa el módulo. Además, se capturarán los requisitos necesarios para la implementación.

Definiciones de los conceptos:

- ✓ **Usuario:** describe a las personas que interactúan con la plataforma.
- ✓ **Plataforma educativa:** representa el entorno donde se gestionan los temas de interfaz de usuario.
- ✓ **Administrador central:** describe al usuario que accede a gestionar un tema y junto con este todos los procesos que lo componen, puede realizar todas las actividades administrativas sobre la gestión de temas.
- ✓ **Usuario de Bachiller:** describe al usuario que generaliza un grupo de usuarios y actividades comunes en diferentes actores, dentro del hiperentorno.
- ✓ **Estudiante:** describe a la persona que cursa estudios para obtener el grado de bachillerato.



- ✓ **Docente:** describe a la persona responsable de educar, guiar y supervisar a los estudiantes en el proceso de enseñanza-aprendizaje.
- ✓ **Tutor:** describe a la persona responsable de guiar y supervisar las actividades de los estudiantes que atiende.
- ✓ **Administrador local:** describe a la persona encargada de realizar las tareas de administración en la plataforma instalada en la escuela.
- ✓ **Director:** describe a la persona encargada de dirigir una escuela, ya sea por ser el propietario o porque se le asignó la tarea. Puede revisar y analizar la información a nivel estudiante, curso, docente y escuela.
- ✓ **Temas:** describe a la estructura y estilo que agrupa el contenido dentro de los módulos de Bachiller.
- ✓ **Tema base:** describe al tema definido dentro de las políticas de la plataforma, con las opciones de presentación específicas que debe cumplir el diseño de la interfaz de Bachiller.
- ✓ **Aplicaciones:** describe el espacio que permite modificar la ubicación que contendrán los Widgets, pequeñas aplicaciones que pueden estar presente dentro de la plataforma.
- ✓ **Propiedades:** describe los elementos que pueden ser configurados dentro de los temas.

2.2 Descripción del sistema propuesto

La Plataforma Educativa Zera cuenta con el subsistema Bachiller, base de la gestión académica, el cual está integrado por temas que son asignados desde el subsistema de Administración, este subsistema contiene un espacio que permite cargar temas a la plataforma y listarlos una vez cargados, pero no permite modificar los temas desde la plataforma, el sistema que se propone facilitará la gestión de los temas de interfaz de usuario, permitiendo la creación, eliminación y edición de las plantillas, además de una vista previa del tema seleccionado y el posicionamiento de los Widgets. A continuación se presenta la estructura que deben contener los temas que se encuentran alojados en el directorio: `web/uploads/themes/tema` en cuestión (Ver Figura 2. Estructura de los temas)

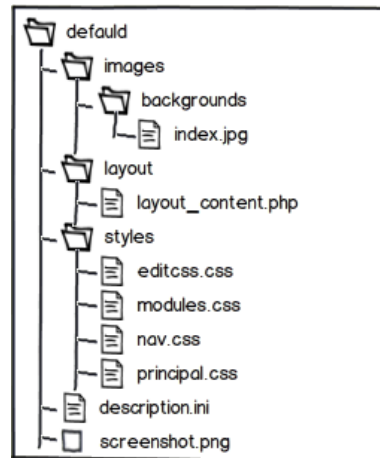


Figura 2. Estructura de los temas

La primera carpeta representa los temas que contendrá la plataforma, siendo “default” el nombre que el usuario le asignará a la hora de crearlo. Este tema contendrá tres carpetas principales images: la cual tendrá una estructura de carpetas, cada una de ellas con las imágenes representativas del tema; layout: contendrá los archivos de configuración de la estructura del tema; styles: tendrá los archivos CSS que son necesarios para la edición y configuración del tema. Además incluirá el archivo description.ini, con el nombre y la descripción del tema y una imagen representativa del mismo.

Dentro de la edición de los temas permitirá modificar las opciones generales, tipografías, tablas, formularios, botones, alertas, barra de navegación, menú desplegable, paginado, layouts, estilos CSS y los estilo de los módulos: Contenido, Tareas, Prácticas, Biblioteca y Docente.

2.3 Especificación de los requerimientos del software

La especificación de requisitos cumple un papel primordial en el proceso de producción de software, enfocando un área fundamental, la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema; de esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas.



2.3.1 Requerimientos funcionales

Los requisitos funcionales (RF) de un software definen lo que el sistema debe hacer, funcionalidades requeridas y restricciones que son aprobadas en mutuo acuerdo con el usuario final. En el siguiente epígrafe se expone un resumen del catálogo de requisitos funcionales de la gestión de temas de interfaz de usuario para el subsistema Bachiller de la Plataforma Educativa Zera.

RF 1 Listar temas de interfaz de usuario: esta funcionalidad brinda la posibilidad de listar los temas que se encuentran cargados en la plataforma y las acciones que se pueden realizar sobre un tema.

RF 2 Eliminar temas de interfaz de usuario: esta funcionalidad permitirá al administrador central eliminar un tema de la plataforma.

RF 3 Crear tema de interfaz de usuario: esta funcionalidad carga el tema base y muestra las opciones a editar dentro de un tema, hace una referencia a los requisitos funcionales de editar opciones de un tema.

Para Editar las opciones de un tema: los requisitos funcionales que se describirán a continuación, formarán parte del RF 4 Editar opciones del tema.

RF 5 Editar opciones generales del tema: permitirá modificar las opciones generales que presenta el tema como pueden ser nombre, descripción, imagen, los colores del fondo del cuerpo del tema, del texto, del capítulo, de los enlaces y enlaces activados. Además permitirá modificar la imagen del banner y del fondo.

RF 6 Editar tipografías: brindará la posibilidad de modificar los estilos de la tipografía en cuanto a tamaño de fuente y familia de la fuente.

RF 7 Editar tablas: permitirá editar las tablas del tema en cuanto a los colores de fondo y borde.

RF 8 Editar formularios: brindará la posibilidad de modificar los formularios del tema en cuanto a los colores de texto, fondo y borde, además del tamaño que contendrá el radio del borde de entrada.

RF 9 Editar botones: permitirá editar los colores de fondo y fondo destacado, de los botones del tema.

RF 10 Editar alertas: permitirá modificar los colores de fondo, texto y borde de las diferentes notificaciones de alerta que presenta el tema (advertencia, error, éxito e información).



RF 11 Editar barra de navegación: permitirá editar los colores del texto, del fondo, de los enlaces y los bordes, del menú de navegación.

RF 12 Editar menú desplegable: permitirá editar los colores del texto, del fondo, de los enlaces y los bordes, del menú desplegable.

RF 13 Editar paginado: permitirá modificar las imágenes de las flechas de navegación del paginado.

RF 14 Editar archivos de configuración de layouts: permitirá modificar la estructura de los layouts que presentan los temas a través del código php.

RF 15 Editar archivos de estilos CSS: permitirá editar el código CSS de los ficheros Principal y Módulos del tema.

RF 16 Editar archivos de estilo CSS del módulo Contenido: permitirá editar el código CSS que presenta el módulo Contenido.

RF 17 Editar imágenes del módulo Contenido: permitirá editar las imágenes que presenta el módulo Contenido.

RF 18 Editar archivos de estilo CSS del módulo Tareas: permitirá editar el código CSS que presenta el módulo Tareas.

RF 19 Editar imágenes del módulo Tareas: permitirá editar las imágenes que presenta el módulo Tareas.

RF 20 Editar archivos de estilo CSS del módulo Prácticas: permitirá editar el código CSS que presenta el módulo Prácticas.

RF 21 Editar archivos de estilo CSS del módulo Biblioteca: permitirá editar el código CSS que presenta el módulo Biblioteca.

RF 22 Editar imágenes del módulo Biblioteca: permitirá editar las imágenes que presenta el módulo Biblioteca.



RF 23 Editar archivos de estilo CSS del módulo Docente: permitirá editar el código CSS que presenta el módulo Docente.

RF 24 Editar imágenes del módulo Docente: permitirá editar las imágenes que presenta el módulo Docente.

RF 25 Ver vista previa: mostrará en una nueva pestaña del navegador el tema con ejemplos de cada uno de los elementos editados (tipografía, tablas, formularios, botones, alertas y paginado), además de los colores e imágenes del fondo, banner y barra de navegación.

RF 26 Posicionar Widgets: brindará la posibilidad de listar los Widgets que se encuentran dentro de cada tema y mostrará las áreas donde se pueden ubicar dichas aplicaciones dentro del tema, estos Widgets podrán ser arrastrados desde la lista y posicionados en el área indicada

2.3.2 Requerimientos no funcionales

Los requerimientos no funcionales (RNF) son propiedades o cualidades que el producto debe cumplir, características que hacen al producto usable, rápido y confiable. Aunque los mismos no aportan funcionalidades propiamente dichas dentro de la aplicación, son de vital importancia para una puesta en marcha exitosa del software, y para lograr que este responda a las expectativas del usuario.

Usabilidad

RNF 1. El sistema debe tener acceso al menú general de navegación desde cualquiera de sus páginas.

RNF 2. Debe contener las rutas de acceso según la navegación como uno de los elementos de usabilidad.

RNF 3. Las imágenes y gráficos deben contener un mensaje de texto flotante mostrando el tipo de recurso al que se refiere.

RNF 4. La edición de las plantillas debe permitir todo tipo de colores para los usuarios con problemas visuales puedan acceder a la información de manera eficiente.

Confiabilidad

RNF 5. Debe restringirse el acceso a los usuarios de acuerdo a las acciones que tienen designadas.

Seguridad de acceso y administración de usuarios: otorgamiento de privilegios y roles, asignación de



perfiles. Los niveles de acceso están determinados por los diferentes roles válidos dentro de la misma, y que se referencia a continuación:

- ✓ Administrador central: este usuario es el autorizado de la administración global del sitio, por lo que se encarga de gestionar los temas de interfaz de usuario para el subsistema Bachiller.
- ✓ Usuario de Bachiller: son los usuarios (estudiantes, profesor, tutor, director y administrador local) encargados de acuerdo a sus funciones de utilizar el sistema creado y visualizar los temas una vez asignados por el administrador central.

RNF 6. Verificación sobre acciones irreversibles (la eliminación de archivos).

Restricciones de diseño

RNF 7. El diseño de las interfaces debe estar en correspondencia con las exigencias de los usuarios y debe ser sugerente para los mismos.

RNF 8. El sistema proporcionará claridad, facilidad de uso y una correcta organización de la información para los usuarios, que permita la interpretación correcta e inequívoca de la misma.

RNF 9. Debe contener un diseño sencillo con poca aglomeración de imágenes y gráficos para que el tiempo de carga del sistema sea mínimo.

Requisitos para la implementación

RNF 10. La aplicación será implementada en el IDE NetBeans 7.1. Se utilizará como marco de trabajo base de desarrollo Symfony 1.4.2. Como servidor web se manejará Apache 2.2.22.

RNF 11. Lenguaje y tecnología de desarrollo: XHTML 1.0, XML 1.0, CSS 3.0, Java Script, Ajax y PHP 5.3.10.

RNF 12. Para el análisis y el diseño de la aplicación debe ser utilizada la metodología RUP, usando el lenguaje de modelación UML 2.1 y como herramienta Visual Paradigm 8.0.

Requisitos de software

Para el cliente:

RNF 13. Las computadoras deben tener instalado el navegador Mozilla Firefox 3.0 o superior, Internet Explorer 7 o superior, Google Chrome 6.0 o superior, Opera 9 o superior, Safari 4.0 o superior.

Para el servidor:

RNF 14. Sistema operativo Linux y servidor Apache v2.2.22.



Requisitos de hardware

Para el servidor:

RNF 15. 1 Procesador CoreDuo, 2GB de RAM, 120 GB de HDD mínimo, aunque puede ser configurable de acuerdo a la cantidad de contenidos que se vayan contratando y banda ancha.

Para el usuario final:

RNF 16. Procesador Pentium II o superior, 1GB de RAM, 20GB de HDD, bocinas, si no cuentan con servidor local: conexión de banda ancha 256 bps como mínimo y si cuenta con servidor local: acceso a red interna.

Requisitos Legales, de Derecho de Autor y otros

RNF 17. Una vez terminado el producto, el sistema debe ser sometido a una evaluación y certificación por parte del cliente del mismo.

Seguridad

RNF 18. Con respecto a la seguridad del código: utilizar ofuscadores de código fuente. Asegurar la plataforma de los ataques de hackers, mediante la escritura de código seguro.

2.4 Presentación del diagrama de caso de uso

A continuación se describirá el actor que interactúa con los casos de uso (CU) y se realizarán sus respectivas descripciones.

2.4.1 Descripción del actor del sistema

Tabla 1. Descripción del actor del sistema

Actor	Descripción
Administrador central	Usuario encargado de administrar los temas de interfaz de usuario para el subsistema Bachiller.

2.4.2 Patrones de casos de uso

Los patrones de casos de uso son comportamientos que deben estar presentes en el sistema, ayudan a describir qué es lo que el sistema debe hacer, su uso y la forma de interactuar con los usuarios. Estos



patrones son utilizados generalmente como plantillas que describen cómo deberían ser estructurados y organizados los casos de uso. Son patrones que capturan mejores prácticas para modelar casos de uso (49).

A continuación se describe el patrón de caso de uso utilizado que ayudó a conformar el diagrama de caso de uso del sistema:

Patrón Concordancia

Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y son expresadas por separado. Tiene dos variantes adición y reuso, la utilizada en el diagrama de caso de uso es adición.

Adición: la subsecuencia común de casos de uso, extiende los casos de uso compartiendo la subsecuencia de acciones. Los otros casos de uso modelan el flujo que será expandido con la subsecuencia. Este patrón es preferible usarlo cuando otros casos de uso se encuentran propiamente completos, o sea, que no requieren de una subsecuencia común de acciones para modelar los usos completos del sistema (49).

Un ejemplo de su uso se muestra en el Anexo 1. Ejemplo del patrón de caso de uso Adición. La subsecuencia común se evidencia en el CU Vista previa, del cual extienden los casos de uso Editar opciones del tema y Crear temas de interfaz de usuario.

2.4.3 Diagrama de caso de uso del sistema

Los casos de uso son el componente clave del modelado. Su propósito es mostrar como un sistema permite a un actor cumplir una meta, ilustrando todos los posibles caminos apropiados que pueden tomar para cumplirla, así como las situaciones que podrían hacerlo fallar. Un diagrama de casos de uso del sistema representa gráficamente las funcionalidades principales del sistema y su interacción con los actores. A continuación se presenta el diagrama de casos de uso del sistema (Ver Figura 3. Diagrama de casos de uso)

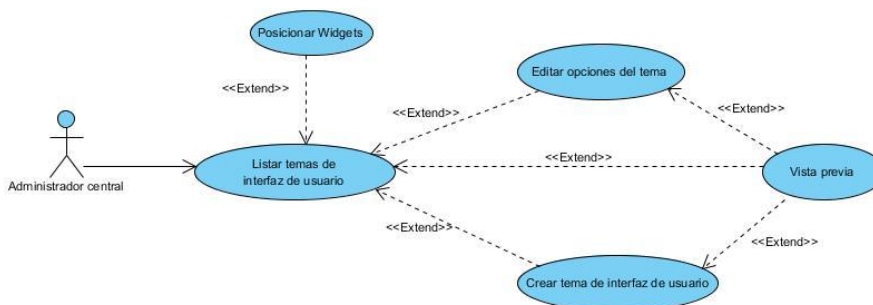


Figura 3. Diagrama de casos de uso

2.4.4 Descripción de los casos de uso

A continuación se presenta un resumen de la descripción del CU Crear temas de interfaz de usuario. Para consultar las descripciones remitirse al Anexo 2. Descripción de casos de uso.

Tabla 2. Descripción del CU Crear temas de interfaz de usuario

CU Crear temas de interfaz de usuario.	
Objetivo	Crear los temas de interfaz de usuario para el subsistema Bachiller.
Actores	Administrador central
Resumen	El caso de uso se inicia cuando el administrador central selecciona la opción que le permite crear un tema. El actor puede realizar cualquier modificación sobre el tema. Si el actor accede a modificar las opciones del tema, el sistema mostrará los elementos a modificar. Si el actor accede a la vista previa del tema, el sistema mostrará el tema con las modificaciones realizadas. Una vez realizados los cambios, guardará las modificaciones, terminando así el caso de uso.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado. Se debe acceder a la página de administración de Materias. Para crear un tema debe haberse listado los temas existentes en la plataforma.
Postcondiciones	Se ha creado el tema de interfaz de usuario por el actor.



Conclusiones del capítulo

En el presente capítulo se llevó a cabo la descripción de los procesos del negocio, obteniéndose el modelo de dominio, el listado con las propiedades que debe cumplir el sistema y las funcionalidades que debe realizar, representando las últimas mediante un diagrama de casos de uso del sistema y finalmente se describen las acciones del actor con los casos de uso que interactúa.



Capítulo III: Análisis y diseño de la propuesta del sistema

Introducción

En el capítulo se realiza el análisis y diseño de las funcionalidades a implementar. Se generan artefactos necesarios que contribuyen a la implementación de dichas funcionalidades, diagramas de clases del análisis y del diseño para cada caso de uso del sistema. Se definen los patrones de diseño que manejan el proceso.

3.1 Modelo del análisis

A continuación se explican los estereotipos presentes en el Diagrama de Clases del Análisis (DCA):

- ✓ **Clase interfaz:** modela la iteración entre el sistema y sus actores.
- ✓ **Clase controladora:** coordina la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.
- ✓ **Clase entidad:** modela la información que posee larga vida y que es a menudo persistente.

3.1.1 Diagrama de clases del análisis

Los diagramas de clases del análisis son la entrada para la etapa de diseño y representan la relación que existe entre clases y casos de uso. A continuación se presenta un ejemplo de DCA (Ver Figura 4. DCA Crear temas de interfaz de usuario), para un análisis detallado remitirse al Anexo 3. Diagramas de Clases del Análisis y al Anexo 4. Diagramas de Interacción del Análisis.

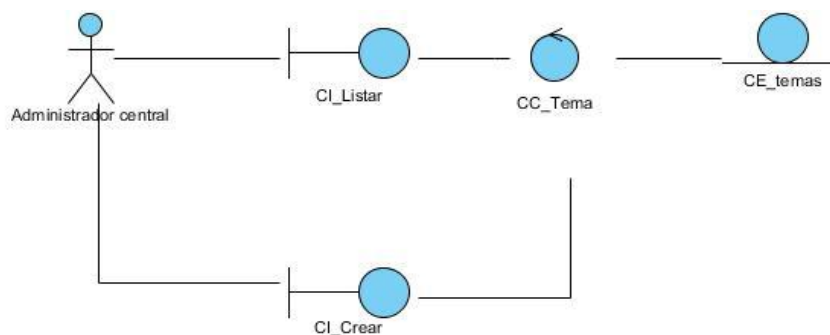


Figura 4. DCA Crear temas de interfaz de usuario



3.2 Patrones arquitectónicos y de diseño en Symfony

Para realizar el modelo del diseño del sistema es necesario comprender las características de los patrones que utiliza Symfony, tanto los que definen la estructura de un sistema como los que proveen información sobre su diseño. A continuación se expondrán los patrones arquitectónicos y de diseño a utilizar en el desarrollo del módulo de gestión de temas de interfaz de usuario para el subsistema Bachiller.

3.2.1 Patrón arquitectónico Modelo-Vista-Controlador

Symfony permite la utilización de varios patrones arquitectónicos en su diseño, el seleccionado para el módulo de gestión de temas es el Modelo Vista Controlador (MVC), el cual es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (50):

- ✓ **Modelo:** maneja los datos y controla todas sus transformaciones, no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo. El modelo está representado por el fichero de configuración de los Widgets y el directorio donde se guardan los estilos de los temas.
- ✓ **Vista:** maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo. Representan las páginas que se visualizan en el directorio `templates/` de la siguiente forma: `nombreSuccess.php` y `_nombreSuccess.php`.
- ✓ **Controlador:** proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, se activa ya sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo. Se evidencia en el controlador frontal cuando redirecciona los pedidos que el usuario hace sobre la vista, al par módulo/acción correspondiente.



3.2.2 Patrones de diseño en Symfony

Para mejorar la calidad de las arquitecturas de software se usan patrones, que capturan existencia, experiencia comprobada en el desarrollo del software y ayudan a promover buenas prácticas de diseño. Symfony, trae incluido por defecto en su arquitectura varios patrones de diseño, que están concebidos para que el programador los aplique en su desarrollo. A continuación se exponen algunos de los patrones utilizados directamente en la solución.

Patrones GoF

Los patrones de diseño tratan los problemas del diseño de software que se repiten y que se presentan en situaciones particulares, con el fin de proponer soluciones a ellas. Son soluciones exitosas a problemas comunes. Existen muchas formas de implementar patrones de diseño, los detalles de las implementaciones se llaman estrategias. Son soluciones simples y elegantes a problemas específicos del diseño de software orientado a objetos (51).

Singleton (Instancia única): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia (52). Un ejemplo de este patrón se evidencia en la clase `sfRouting` (método `getInstance`) que utiliza el controlador frontal (`sfWebFrontController`) y se encarga de enrutar todas las peticiones que se hagan a la aplicación. El singleton `sfRouting` precisa otros métodos muy útiles para la gestión manual de las rutas: `ClearRoutes()`, `hasRoutes()`, `getRoutesByName()`.

Decorator (Decorador): añade funcionalidad a una clase dinámicamente. Evita efecto cascada y contar con una jerarquía de clases extensa (52). Este patrón se ejemplifica con la utilización del archivo nombrado `layout.php` que contiene el layout de la página, este archivo, conocido también como plantilla global, guarda el código HTML que es usual en todas las páginas del sistema, y es utilizado para crear nuevas funcionalidades.

Patrones GRASP

Los patrones GRASP (Patrones para Asignar Responsabilidades), en sistema orientado a objetos se componen de objetos que envían mensajes a otros objetos para que lleven a cabo las operaciones requeridas. Los diagramas de interacción describen gráficamente estas operaciones, a partir de los objetos en interacción, que se responsabilizan de una actividad determinada (53).



Experto: asigna una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad, es un patrón que se utiliza al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Un ejemplo de este patrón se evidencia en la responsabilidad que le asigna la clase Actions a la clase themeUpdate, encarga de gestionar las acciones sobre los temas.

Alta Cohesión: es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. Con el uso de este patrón mejoran la claridad y la facilidad con que se presenta el diseño. Un ejemplo de ello es la clase Actions, la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades.

Bajo Acoplamiento: es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. Este patrón estimula la asignación de una responsabilidad, de modo que su colocación no incremente el acoplamiento, para que no produzca los resultados negativos propios de un alto acoplamiento. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables. La Actions hereda únicamente de sfActions para alcanzar un mínimo acoplamiento de clases, este es un ejemplo del patrón bajo acoplamiento.

Creador: asigna responsabilidades relacionadas con la creación de objetos, cuyo propósito fundamental es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Brinda un soporte a un bajo acoplamiento lo que supone menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización. Un ejemplo de este patrón es el Actions que se encarga de asignar la responsabilidad de crear los objetos a las clases themeUpdate y widgetsUpdate.

Controlador: este patrón ofrece una guía para tomar decisiones apropiadas que generalmente se aceptan. Garantiza que la empresa o los procesos de dominio sean manejados por la capa de los objetos del dominio y no por la de la interfaz. Este patrón se evidencia en las peticiones web, que se realizan por un controlador frontal (sfActions), el cual es el punto de entrada único de toda la aplicación en un entorno determinado.



3.3 Modelo de diseño

3.3.1 Diagrama de clases del diseño

Los Diagramas de Clases del Diseño (DCD) describen gráficamente las especificaciones de las clases y las interfaces en una aplicación. Son empleados para representar las relaciones que se establecen entre las clases (54). A continuación se presenta un ejemplo de DCD (Ver Figura 5. DCD Crear temas de interfaz de usuario), para el análisis detallado de los diagramas de diseño consultar el Anexo 5. Diagramas de Clases del Diseño.

La carpeta themes recoge todos los archivos encargados de guardar las configuraciones realizadas a cada tema, por lo que corresponde a la clase del modelo en los diagramas de diseño.

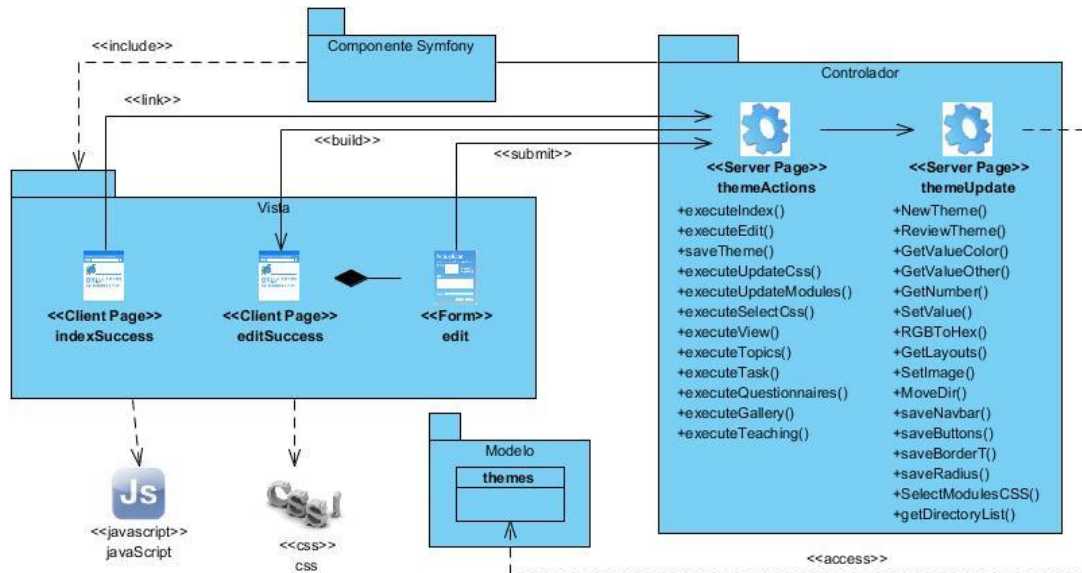


Figura 5. DCD Crear temas de interfaz de usuario

3.4 Modelos de datos

En este trabajo no es necesaria la realización de un modelo de datos debido a que no se utilizan tablas de la base datos, el sistema de temas utiliza un directorio llamado theme donde se almacena toda la información referente a los temas y un fichero yml donde se guarda toda la configuración de los Widgets.



Conclusiones del capítulo

En el capítulo se obtuvo como resultado los artefactos que se generan durante el flujo de trabajo análisis y diseño, entre ellos el modelo del análisis, donde se construyen los diagramas de clases y los diagramas de interacción del análisis, el modelo de diseño, donde se confeccionan los diagramas de clases del diseño. Se describieron y tuvieron en cuenta los patrones que aporta el framework Symfony para desarrollar el sistema propuesto.



Capítulo IV: Implementación y pruebas de las funcionalidades

Introducción

En el capítulo se especifican los estándares de codificación utilizados, cómo los elementos del modelo del diseño se implementan en términos de componentes, y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. Para efectuar pruebas a los componentes desarrollados, se realiza la evaluación de la calidad del producto, mediante la validación eficiente de los requisitos.

4.1 Estándares de codificación

Para el desarrollo de cualquier aplicación dentro de un proyecto es necesario establecer un estándar de codificación para asegurar que todos los programadores trabajen de forma coordinada. Cuando la aplicación incorpore código fuente previo, o cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación permitirá operar con la base de código existente. A continuación se presentan los estándares utilizados en el desarrollo del módulo de gestión de temas de interfaz de usuario:

Convenio de nombres

Los nombres deben ser cortos y descriptivos, facilitando el entendimiento del código generado. Se deben utilizar comentarios en todos los casos que sean normados y en caso de que el programador lo considere necesario. Para los nombres se establecen las siguientes reglas:

- ✓ Funciones y métodos

Las funciones y métodos se deben nombrar usando el formato notación CamelCase.

- ✓ Constantes

El código de Symfony no incluye ninguna constante, ya que utiliza su propio objeto para almacenar la configuración, llamado sfConfig, y que reemplaza a las constantes.

```
// Symfony utiliza el objeto sfConfig
$upload = sfConfig::get('sf_upload_dir');
$root = sfConfig::get('sf_root_dir');
```

- ✓ Clases

Los nombres de las clases contienen el formato UpperCamelCase.



- ✓ Atributos y métodos

Los atributos y los métodos se deben nombrar siguiendo el formato lowerCamelCase. Ejemplos:

```
saveButtons()
```

```
saveRadius()
```

- ✓ Módulos

Se deben escribir en minúsculas, sin guión bajo entre palabras.

- ✓ Acciones

Las acciones deben comenzar con la palabra execute seguido con el nombre de la acción en letra inicial mayúscula. Ejemplo:

```
executeIndex()
```

- ✓ Plantillas

El nombre debe comenzar con el nombre de la acción en minúscula seguido de la palabra Success. Por ejemplo:

```
indexSuccess.php
```

- ✓ Ficheros

Los nombres de los archivos que contienen la declaración de una clase tienen extensión .class.php. Los nombres de los archivos de configuración del proyecto, aplicaciones y módulos tienen extensión .yaml

Operadores

Todos los operadores binarios, tales como +, -, =, !=, ==, >, etc. deben llevar un espacio antes y después del operador. Por ejemplo, una asignación debe tener el formato \$value = \$theme; en vez de \$value=\$theme. Los operadores unarios, tales como ++, no deben tener espacio entre el operador y el operando.

Escribir CSS válido

Todos los documentos deben tener CSS válido, preferiblemente CSS 3.0.

La terminología usada es:

```
selector {  
  propiedad: valor; }
```



Los selectores deben estar seguidos con una llave abierta en la próxima línea. Luego de las propiedades debe terminar con una llave cerrada con la misma sangría de la llave abierta. Una línea en blanco debe existir entre cada bloque de selectores.

Java Script

- ✓ Alineación y espacios en blanco

Se deben utilizar 2 espacios como unidad de alineación en vez de utilizar la tecla TAB (8 espacios).

- ✓ Variables

Todas las variables deben ser declaradas con variables antes de que se utilicen y sólo se deben declarar una vez. Hacer esto hace que el programa sea más fácil de leer y sea más fácil detectar las variables no declaradas que puedan convertirse en globales implícitas.

Las variables no deben definirse en el ámbito global, se deben tratar de definir dentro de una función. Todas las variables deben ser declaradas al principio de una función.

- ✓ Constantes

Las constantes predefinidas deben tener el formato lowerCamelCased. A diferencia de los estándares de PHP, debe usar la versión minúscula de true, false, y null ya que las versiones mayúsculas no son válidas en Java Script.

- ✓ Ubicación del código Java Script

El código Java Script no debe ser embebido en el HTML siempre que sea posible, ya que añade mucho peso a la página sin posibilidades para mitigación y compresión.

4.2 Flujo de trabajo de implementación

El flujo de trabajo Implementación tiene como objetivo implementar las clases del diseño en forma de componentes, asignar los componentes a los nodos, probar los componentes individualmente e integrar dichos componentes en un sistema ejecutable.

4.2.1 Diagrama de componentes

Los Diagramas de Componentes (DC), son la colección de componentes y subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, entre otros que serán necesarios para la implantación del sistema. Muestran cómo están distribuidos según el patrón



arquitectónico Modelo-Vista-Controlador que utiliza Symfony. A continuación se presenta el DC general que describe los componentes a utilizar en el desarrollo de la implementación.

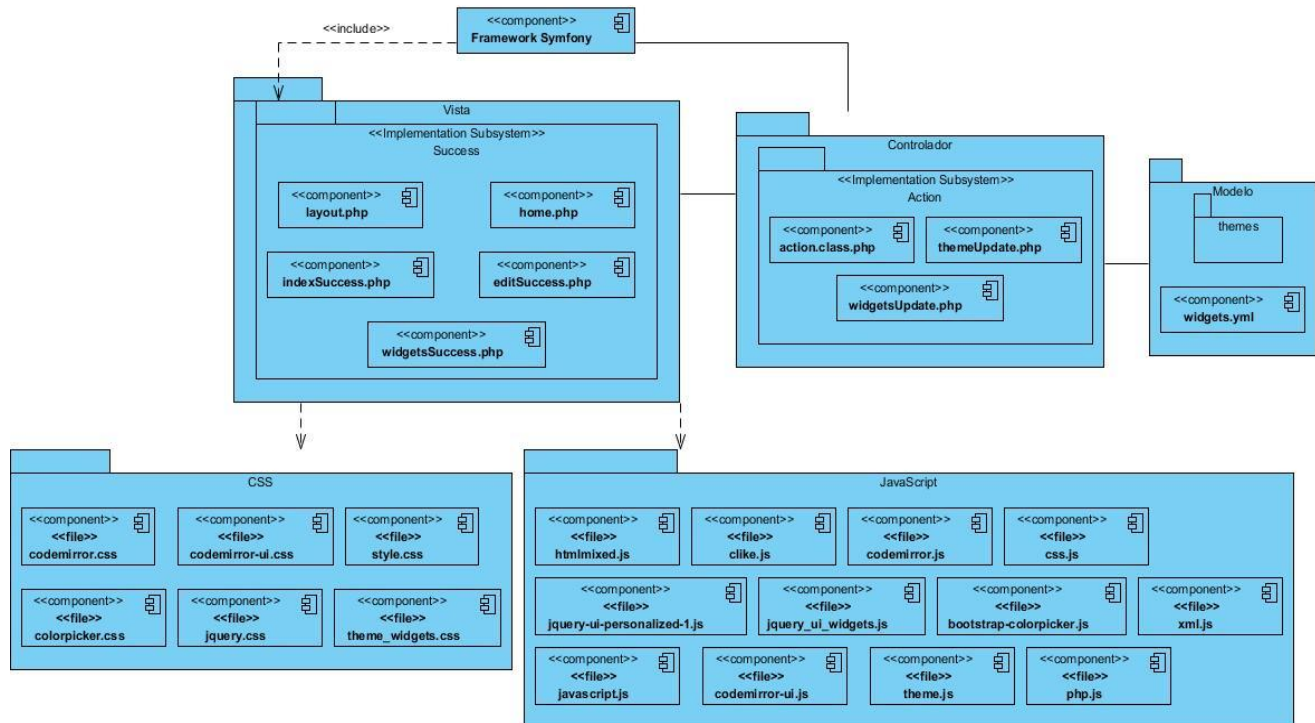


Figura 6. Diagrama de componentes general

4.2.2 Diagrama de despliegue

Otro de los artefactos generados en el flujo de trabajo de implementación es el diagrama de despliegue (Ver Figura 7. Diagrama de despliegue), donde se representan los nodos que componen el sistema, además de especificar los protocolos que utilizan para comunicarse. Como diagrama de despliegue para el módulo de gestión de temas de interfaz de usuario se hace uso del diagrama definido en la arquitectura de la Plataforma Educativa Zera, debido a que el módulo y el subsistema de gestión de materias son parte de la misma y por tanto, están alojados en el mismo servidor y utilizan la misma base de datos.

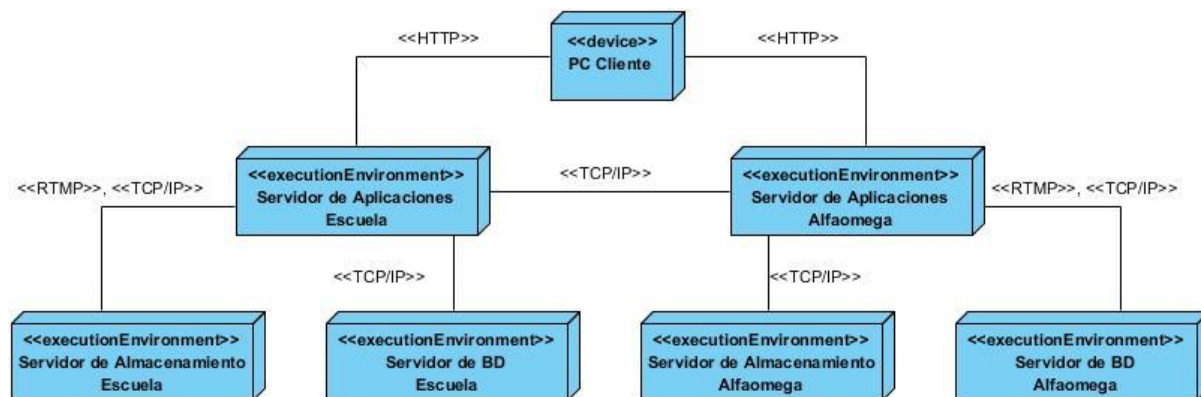


Figura 7. Diagrama de despliegue

4.3 Pruebas de software

Este flujo de trabajo es el encargado de evaluar la calidad del producto que se está desarrollando y presenta como objetivos principales encontrar y documentar defectos en la calidad del software. Provee la validación de los supuestos realizados en el diseño y especificación de requisitos por medio de demostraciones concretas y verifica las funciones del producto de software según lo diseñado.

4.3.1 Niveles de Prueba

Las pruebas pueden realizarse en diferentes escenarios o niveles de trabajo dependiendo del objetivo de los mismos. Tienen un enfoque incremental, inicia con las pruebas de unidades individuales del programa, pasa a pruebas diseñadas para facilitar la integración de las unidades y culmina con las pruebas que se realizan sobre el sistema construido.

Pruebas unitarias

Estas pruebas fueron realizadas por el desarrollador a medida que se implementaba la solución para luego conformar sus partes con el resto de la plataforma, tuvo como objetivo principal aislar el módulo y mostrar su funcionamiento de forma individual, además de asegurar el correcto funcionamiento de las interfaces de usuario.



Pruebas de integración

Las pruebas de integración se realizaron una vez aprobadas las pruebas unitarias. Consisten en las pruebas de software donde el módulo de gestión de temas es combinado y probado junto a los demás módulos de la plataforma.

Pruebas del sistema

Las pruebas del sistema se realizaron sobre cada componente del módulo, verificando el cumplimiento de los requisitos funcionales y no funcionales establecidos. Para el desarrollo de estas pruebas se utilizaron casos de pruebas que permiten obtener el flujo adecuado de las funcionalidades.

4.3.2 Métodos de Prueba

RUP define dos métodos de pruebas: caja negra y caja blanca, a continuación se expondrán sus características principales haciendo énfasis en las de caja negra, pues se diseñan los casos de prueba (CP) que permiten comprobar la validez de las funcionalidades de acuerdo a las acciones del usuario, entradas y respuestas del sistema.

Pruebas de Caja Blanca

Las pruebas de Caja Blanca se realizan a medida que se desarrolla el módulo de gestión de temas comprobando el funcionamiento adecuado del código y cumpliendo con los estándares establecidos para la implementación de los componentes dentro de la plataforma. No se utilizan métodos de Caja Blanca sobre el código, ya que este no presenta complejidad en sus acciones y puede ser de fácil comprensión para los desarrolladores.

Pruebas de Caja Negra

Las pruebas de Caja Negra no se realizan como alternativa a las pruebas de Caja Blanca, sino para intentar descubrir diferentes tipos de errores que se llevan a cabo sobre la interfaz del software. Mediante la descripción de CP se demuestra que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto. A continuación se expone el diseño de CP que ayuda a la realización de las pruebas de Caja Negra.



Diseño de Casos de Prueba

Para el diseño de los casos de pruebas es necesario un número de datos que ayuden a la ejecución de todas sus variantes, pueden ser datos válidos o inválidos para el programa según lo que se desea realizar. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos, a fin de verificar que el programa se ejecute en perfectas condiciones (55).

A continuación se presenta el diseño de CP Listar temas de interfaz de usuario. Para consultar los demás CP dirigirse al Anexo 6. Diseños de Casos de Prueba.

Tabla 3. CP Listar temas de interfaz de usuario

Diseño de Casos de Prueba: CP Listar temas de interfaz de usuario			
Descripción General			
<p>El caso de uso se inicia cuando el actor accede a la opción de Listar temas de interfaz de usuario. El sistema muestra los temas que se encuentran cargados en la plataforma. Permite consultar la estructura de un tema, así como realizar cualquier acción sobre él. En caso de acceder a eliminar un tema el sistema permitirá eliminar los que no cumplan con los requisitos del usuario. El caso de uso termina.</p>			
Condiciones de Ejecución			
<p>Debe haberse generado el escritorio de trabajo del usuario autenticado.</p>			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Selecciona Listar los temas.	El actor selecciona la opción de listar los temas de interfaz de usuario.	<p>Muestra un listado de los temas que se encuentran cargados en la plataforma con los datos:</p> <ul style="list-style-type: none"> • Título. • Descripción. • Dirección del tema. • Fecha de publicación. <p>Permite:</p> <ul style="list-style-type: none"> • Crear tema. <p>Y de cada tema permite:</p> <ul style="list-style-type: none"> • Editar • Eliminar • Ver vista previa • Posicionar Widgets 	Módulo Materia/Temas



EC 1.2	El actor selecciona la opción de crear un nuevo tema.	Brinda la posibilidad de crear un tema. <u>Ver CP "Crear tema de interfaz de usuario"</u> .	Módulo Materia/Temas/Crear nuevo tema
EC 1.3	El actor selecciona la opción de editar un tema.	Brinda la posibilidad de editar un tema. <u>Ver CP "Editar opciones del tema"</u> .	Módulo Materia/Temas/Editar
EC 1.4	El actor selecciona la opción de eliminar un tema.	Brinda la posibilidad de eliminar un tema. <u>Ver SC 1: "Eliminar tema de interfaz de usuario"</u> .	Módulo Materia/Temas/Eliminar
EC 1.5	El actor selecciona la opción de ver un tema.	Brinda la posibilidad de visualizar un tema. <u>Ver CP "Vista previa"</u> .	Módulo Materia/Temas/Vista Previa
EC 1.6	El actor selecciona la opción de posicionar los Widgets de los temas.	Brinda la posibilidad de posicionar los Widgets de los tema. <u>Ver CU "Posicionar Widgets"</u> .	Módulo Materia/Temas/Posicionar Widgets
Sección 2: Eliminar temas de interfaz de usuario.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 2.1	El actor selecciona la opción de eliminar un tema de la plataforma.	Muestra un mensaje de confirmación. Y permite: • Aceptar • Cancelar	Módulo Materia/Temas/Eliminar
EC 2.2	El actor selecciona la opción de aceptar.	Muestra un mensaje de información. Elimina el tema. Muestra el listado de temas actualizado.	Módulo Materia/Temas/Elimina/Aceptar



EC 2.3 Selecciona cancelar.	El actor selecciona cancelar la operación.	Regresa a la vista anterior.	Módulo Materia/Temas/Elimina/Cancelar
-----------------------------------	--	------------------------------	--

4.3.3 Resultados obtenidos

Se realizaron pruebas unitarias al código durante el desarrollo de las funcionalidades del módulo de gestión de temas de interfaz de usuario para comprobar su correcta implementación. Estas pruebas fueron realizadas por el desarrollador, no fueron planificadas y los resultados no se registraron pues se realizaron conjuntamente con el desarrollo de la solución.

Se desarrollaron pruebas de integración para validar la solución y comprobar su funcionamiento, además de verificar la relación de las funcionalidades implementadas con el resto de los módulos de la plataforma.

Dentro del método de Caja Negra se utiliza el criterio de partición de equivalencia que divide el campo de entrada del módulo en clases de datos de los que se derivan los CP. La partición equivalente se dirige a una definición de CP que descubre clases de errores, reduciendo así el número de CP que hay que desarrollar. A continuación se presentan los resultados asociados a las diferentes iteraciones de pruebas realizadas.

Tabla 4. Resultados de las pruebas (1era Iteración)

Casos de Prueba	No conformidades			
	Alta	Media	Baja	Resueltas
Listar temas de interfaz de usuario	1	2	3	4
Crear temas de interfaz de usuario		3	5	8
Editar opciones del tema		3	5	8
Vista previa		1	2	3
Posicionar Widgets	2		3	3



Tabla 5. Resultados de las pruebas (2da Iteración)

Casos de Prueba	No conformidades			
	Alta	Media	Baja	Resueltas
Listar temas de interfaz de usuario			1	1
Crear temas de interfaz de usuario		3	1	4
Editar opciones del tema		3	1	4
Vista previa		2		2
Posicionar Widgets	2		3	5

Conclusiones del capítulo

En este capítulo se obtuvo el modelo de implementación, donde se describe como los elementos del modelo del diseño se implementan en términos de componentes y como estos son agrupados en subsistemas de implementación para una mayor organización, además se realiza el modelo de despliegue que define la arquitectura física del sistema. Se validaron las funcionalidades implementadas de acuerdo a los requisitos especificados y se conformaron los diseños de casos de prueba como artefacto principal de este flujo de trabajo. Se realizaron pruebas unitarias y de integración arrojando no conformidades, que fueron corregidas en cada iteración.



Conclusiones

Con la realización del presente trabajo se da cumplimiento a los objetivos trazados, obteniéndose como principales resultados las funcionalidades que permiten la gestión de temas de interfaz de usuario de forma eficiente, además se generaron los artefactos que dan cumplimiento a cada uno de los flujos de trabajo que propone RUP. Se identificaron las tendencias actuales de los sistemas que permiten la gestión de temas, a partir del análisis del estado del arte realizado. Se generó la documentación necesaria para la implementación del módulo de gestión de temas de interfaz de usuario, a partir de los artefactos obtenidos durante el flujo de trabajo análisis y diseño. Se permitió a los administradores gestionar los temas de interfaz de usuario a partir de las funcionalidades implementadas. Se validó la propuesta de solución a través de las pruebas realizadas al módulo de gestión de temas de interfaz de usuario.



Recomendaciones

Luego de haber apreciado los resultados obtenidos y la experiencia adquirida durante la realización de la investigación se recomienda perfeccionar el módulo de gestión de temas agregando:

- ✓ Un espacio que permita a los usuarios de Bachiller editar el tema en cuanto a las opciones generales de diseño.
- ✓ La creación de temas personalizados por usuario y escuela, teniendo en cuenta las características de cada uno de ellos.
- ✓ La posibilidad de exportar el tema creado.



Referencias bibliográficas

1. **Rautherberg, Matthias, Menozzi, Marino y Wesson, Janet.** *Human Computer Interaction*. Amsterdam : Nieuwe Hemweb 6B, 2003. ISBN 58603.
2. **Torre, Anibal de la.** *Web Educativa 2.0. Revista Electrónica de Tecnología Educativa*. [Revista Electrónica] Palma del Río : Edutec, Edutec, 2006. Vol. 20. 20.
3. **Balmaseda, Greilan García.** Diseño de interfaz de usuario para software educativo. *Diseño de interfaz de usuario para software educativo*. [En línea] 16 de Febrero de 2012. [Citado el: 1 de Junio de 2013.] <http://publicaciones.uci.cu/index.php/SC/article/view/551/509>.
4. **FORTES.** FORTES - Centro de Tecnologías para la Formación. *FORTES - Centro de Tecnologías para la Formación*. [En línea] Redmine, 2011. [Citado el: 3 de Junio de 2013.] <http://gespro.fortes.prod.uci.cu>.
5. **Real Academia de la Lengua Española.** Diccionario de la Real Academia de la Lengua Española. *Diccionario de la Real Academia de la Lengua Española*. [En línea] 2009. [Citado el: 16 de Mayo de 2013.] <http://www.rae.es>.
6. **El Mundo.es.** Diccionarios El Mundo. *Diccionarios El Mundo*. [En línea] 4 de Febrero de 2008. [Citado el: 10 de Mayo de 2013.] http://diccionarios.elmundo.es/diccionarios/cgi/lee_diccionario.html?busca=interfaz.
7. **Glosarium.com.** Diccionario Informático. *Diccionario Informático*. [En línea] 20 de Enero de 2008. [Citado el: 10 de Mayo de 2013.] <http://www.glosarium.com/list/14/5,l,,xhtml>.
8. **Shnedeirman, Ben.** *El diseño de la interfaz de usuario: estrategias para una eficaz interacción hombre-máquina*. 2005.
9. **Aimacaña Toledo, Carlos.** Interfaz de usuario. *Interfaz de usuario*. [En línea] 2000. [Citado el: 26 de Octubre de 2012.] <http://www.monografias.com/trabajos6/inus/inus.shtml>.



10. **Ferré Grau, Xavier.** Principios Básicos de Usabilidad para Ingenieros Software. *Principios Básicos de Usabilidad para Ingenieros Software*. [En línea] 2010. [Citado el: 10 de Noviembre de 2012.] <http://www.willydev.net/Descargas/Articulos/General/usabilidad.pdf>.
11. **autores, Colectivo de.** Definición de accesibilidad. *Definición de accesibilidad*. [En línea] 2013. [Citado el: 10 de Noviembre de 2012.] <http://definicion.de/accesibilidad>.
12. **Frías Castillo, Amparo.** Los servicios personalizados de información de actualidad de los medios de comunicación españoles a través de Internet. *Los servicios personalizados de información de actualidad de los medios de comunicación españoles a través de Internet*. [En línea] 16 de Diciembre de 2006. [Citado el: 12 de Noviembre de 2012.] www.ub.edu/bid/17frias2.htm.
13. **Ayuda de WebSphere Portal.** Gestión de Temas y Skin. *Gestión de Temas y Skin*. [En línea] 2009. [Citado el: 31 de Mayo de 2013.] http://www.contaduria.gov.co/wps/iehs/index.jsp?topic=/com.ibm.wp.admin.help/admin/h_main_themes_skins.html.
14. **González, Cristóbal.** Los mejores temas para Drupal. *Los mejores temas para Drupal*. [En línea] 2010. [Citado el: 31 de Mayo de 2013.] <http://www.mediaglobe.es/es/blog/los-mejores-temas-para-drupal>.
15. *Entornos Formativos en el ciberespacio: las plataformas educativas.* **Covadonga, Alonso López y Seré, Arlette.** 556459, s.l. : Base de Datos ISOC, 2004, Vol. 17. ISSN: 1135-867X.
16. **Herrera Batista, Miguel Angel y Latapie Venegas, Imelda.** Diseñando para la educación. *Diseñando para la educación*. [En línea] 9 de Marzo de 2010. [Citado el: 17 de Enero de 2013.] http://www.nosolousabilidad.com/articulos/disenio_educacion.htm.
17. **Comunidad de usuarios de Drupal.** Drupal. *Drupal*. [En línea] 11 de Abril de 2005. [Citado el: 21 de Noviembre de 2012.] <http://drupal.org/es/drupal>.



18. **Cantón, Alejandro Castillo.** Como crear un Sub-Theme paso a paso con Drupal. *Como crear un Sub-Theme paso a paso con Drupal.* [En línea] 15 de Octubre de 2011. [Citado el: 18 de Diciembre de 2012.] <http://www.siguaraya.com/archives/382>.
19. **D'Agostino, Mariano.** El sistema de Themes en Drupal. *El sistema de Themes en Drupal.* [En línea] 14 de Diciembre de 2009. [Citado el: 18 de Diciembre de 2012.] http://cuencodigital.com/articulos/el_sistema_de_themes_en_drupal.html.
20. **Saorin Perez, Tomas.** Guía básica de gestión de contenidos web con Drupal: instalación, configuración y extensión. *Guía básica de gestión de contenidos web con Drupal: instalación, configuración y extensión.* [En línea] 7 de Febrero de 2009. [Citado el: 18 de Diciembre de 2012.] <http://digitum.um.es/xmlui/handle/10201/3300>.
21. **Zamora, Marcelo.** Wordpress: Herramienta para manejos de contenidos y blogs. *Wordpress: Herramienta para manejos de contenidos y blogs.* [En línea] 21 de Noviembre de 2006. [Citado el: 22 de Noviembre de 2012.] <http://www.maestrosdelweb.com/editorial/wordpress>.
22. **Gil, Cristian.** Como crear un theme para WordPress. *Como crear un theme para WordPress.* [En línea] 2009. [Citado el: 18 de Diciembre de 2012.] <http://es.scribd.com/doc/32618395/Tutorial-Crear-Theme-Wordpress>.
23. **autores, Colectivo de.** Joomla. ¿Qué es Joomla? *Joomla. ¿Qué es Joomla?* [En línea] 2005. [Citado el: 29 de Noviembre de 2012.] <http://www.joomla.org/about-joomla.html>.
24. **González, Mary Peñalver.** Principales características de Moodle. [En línea] 6 de Abril. [Citado el: 18 de Diciembre de 2012.] <http://www.slideshare.net/MaryPealverGonzlez/principales-caracteristicas-de-moodle>.
25. **Henrick, Gavin.** Temas en Moodle. *Temas en Moodle.* [En línea] 25 de Abril de 2013. [Citado el: 18 de Diciembre de 2012.] <http://docs.moodle.org/all/es/Temas>.
26. **Figueroa, Roberto G. y Cabrera, Armando A.** Metodologías tradicionales vs Metodologías ágiles. *Metodologías tradicionales vs Metodologías ágiles.* 2007.



27. **Kruchten, Philippe.** Rational Unified Process, An Introduction. *Rational Unified Process, An Introduction*. s.l. : Third Edition, 2003.
28. **Gómez, Juan Pablo.** RUP (Proceso Unificado de Rational). *RUP (Proceso Unificado de Rational)*. [En línea] 2009. [Citado el: 28 de Noviembre de 2012.] <http://www.scribd.com/doc/7844685/CONCEPTOS-DE-RUP>.
29. **Sitio oficial de W3c.** Guía breve de XHTML. *Guía breve de XHTML*. [En línea] 2009. [Citado el: 31 de Mayo de 2013.] <http://w3c.es/Divulgacion/GuiasBreves/XHTML>.
30. **Abril., José M.** Observatorio Industrial de Electrónica, Tecnologías de la Información y Telecomunicaciones. *Observatorio Industrial de Electrónica, Tecnologías de la Información y Telecomunicaciones*. [En línea] AMETIC, 1 de Junio de 2011. [Citado el: 13 de Diciembre de 2012.] <http://www.ametic.es/es/Inicio/actualidad/salaprensa/noticias/observatorio-industrial-de-electronica-tecnologias-de-la-informacion-y-telecomunicaciones>.
31. **autores, Colectivo de.** Introducción a CSS. *Introducción a CSS*. [En línea] LibrosWeb, 2008. [Citado el: 22 de Enero de 2013.] <http://www.librosweb.es/css/>.
32. **Pérez, Ilario.** El Código. *El Código*. [En línea] 2008. [Citado el: 28 de Noviembre de 2012.] <http://www.elcodigo.net/tutoriales/javascript/javascript1.html>.
33. **Zaera Avellón, Ivan.** Introducción a la Tecnología Ajax. [En línea] Febrero de 2006. [Citado el: 14 de Enero de 2013.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=beginAjax>.
34. **autores, Colectivo de.** Programación en castellano. *Programación en castellano*. [En línea] 2008. [Citado el: 22 de Enero de 2013.] <http://www.programacion.net/php/>.
35. **W3c España.** Guía Breve de Tecnologías XML. *Guía Breve de Tecnologías XML*. [En línea] 2009. [Citado el: 31 de Mayo de 2013.] <http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>.



36. **Angel Alvarez, Miguel.** Introducción a JQuery. *Introducción a JQuery*. [En línea] Marzo de 2009. [Citado el: 26 de Enero de 2013.] <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>.
37. **Romero, Roger.** Frameworks de ccs para el desarrollo web. *Frameworks de ccs para el desarrollo web*. [En línea] 2011. [Citado el: 10 de Diciembre de 2012.] <http://www.regoremor.com/desarrollo-web/css/frameworks-de-css-para-el-desarrollo-web/>.
38. **Angel Alvares, Miguel.** Listado de distintos frameworks CSS. *Listado de distintos frameworks CSS*. [En línea] Junio de 2009. [Citado el: 2 de Febrero de 2013.] <http://www.desarrolloweb.com/articulos/listado-frameworks-css.html>.
39. **Pedraza Araque, Nancy y Berdugo Torres, Martha.** Un estado del arte del desarrollo de ambientes web instruccionales para lenguas y otras áreas. *Un estado del arte del desarrollo de ambientes web instruccionales para lenguas y otras áreas*. [En línea] Noviembre de 2005. [Citado el: 11 de Febrero de 2013.] <http://www.google.com/cu/url?sa=t&rct=j&q=plataformas+web%2Bestado+del+arte&source=web&cd=4&ved=0CEAQFjAD&url=http%3A%2F%2Frevistalenguaje.univalle.edu.co%2Findex.php%3Fseccion%3DREVISTA%26articuloCompleto%3D140%26download%3D1&ei=-xvwUNH2OMiw0QGT7oHwCg&us>.
40. **autores, Colectivo de.** Introducing Bootstrap. *Introducing Bootstrap*. [En línea] [Citado el: 10 de Febrero de 2013.] <http://twitter.github.com/bootstrap/>.
41. —. Symphony la guía definitiva. *Symfony la guía definitiva*. [En línea] LibrosWeb, 2013. [Citado el: 20 de Febrero de 2013.] http://www.librosweb.es/symfony/capitulo_1/symfony_en_pocas_palabras.html.
42. **Colectivo de autores de Ciberaula.** Una Introducción a Apache. *Una Introducción a Apache*. [En línea] 2010. [Citado el: 22 de Enero de 2013.] http://linux.ciberaula.com/articulo/linux_apache_intro.



43. **NetBeans.** NetBeans. *NetBeans*. [En línea] 2008. [Citado el: 28 de Noviembre de 2012.] <http://www.netbeans.org>.
44. **Deseta, Leonardo.** NetBeans. *NetBeans*. [En línea] 2011. [Citado el: 28 de Noviembre de 2012.] <http://www.dosideas.com/wiki/NetBeans>.
45. **Kruchten, Philippe.** *Visual-Paradigm*. Canada : Addison-Wesley Professional, 2003. ISBN-10: 0321197704.
46. **C., Cristina.** Balsamiq Mockups como herramienta de prototipado. *Balsamiq Mockups como herramienta de prototipado*. [En línea] 19 de Octubre de 2010. [Citado el: 23 de Enero de 2013.] <http://www.blogtrw.com/2010/10/balsamiq-mockups-como-herramienta-de-prototipado/>.
47. **Schweikert, Raphael.** Sabberworm. *Sabberworm*. [En línea] 2011. [Citado el: 12 de Febrero de 2013.] <https://github.com/sabberworm/PHP-CSS-Parser#readme>.
48. **Enrique Hernández, Orallo.** El Lenguaje Unificado de Modelado(UML). *El Lenguaje Unificado de Modelado(UML)*. [En línea] 2006. [Citado el: 02 de Marzo de 2013.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
49. **Cuesta Rodríguez, Saúl.** Patrones de Casos de Uso. *Patrones de Casos de Uso*. [En línea] Noviembre de 2007. [Citado el: 22 de Enero de 2013.] <http://sg.com.mx/content/view/510>.
50. **Catalani, Exequiel.** Arquitectura Modelo-Vista-Controlador. *Arquitectura Modelo-Vista-Controlador*. [En línea] 20 de Agosto de 2007. [Citado el: 16 de Febrero de 2013.] <http://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador/>.
51. **Juárez, Maximiliano.** Patrones de diseño de GOF. *Patrones de diseño de GOF*. [En línea] 9 de Junio de 2011. [Citado el: 16 de Febrero de 2013.] <http://migranitodejava.blogspot.com/2011/05/patrones-de-diseno-de-gof.html>.
52. **Pérez Mariñán, Pablo.** Patrones de Diseño (Design Patterns). *Patrones de Diseño (Design Patterns)*. [En línea] [Citado el: 26 de Febrero de 2013.] http://www.ecured.cu/index.php/Patrones_Gof.



53. **Saavedra, Jorge.** PATRONES GRASP (Patrones de Software para la asignación General de Responsabilidad).Parte II. *PATRONES GRASP (Patrones de Software para la asignación General de Responsabilidad).Parte II.* [En línea] 8 de Mayo de 2007. [Citado el: 26 de Febrero de 2013.] <http://jorgesaavedra.wordpress.com/category/patrones-grasp/>.
54. **Larman, Craig.** Diagramas de Clases Cap1. *Diagramas de Clases Cap1.* [En línea] 2011. [Citado el: 26 de Febrero de 2013.] <http://www.slideshare.net/jpbthames/diagramas-de-clases>.
55. *Gestión de las Pruebas Funcionales.* **Lamanca, Beatriz Pérez.** 4, Montevideo, Uruguay : Centro de Ensayos de Software, 2007, Vol. 1. ISO/IEC.



Glosario de Términos

Educación a distancia: Modalidad educativa en la que los estudiantes no necesitan asistir físicamente a un centro educativo, pueden recibir la información mediante la web.

CSS: Mecanismo simple que describe cómo se muestra la información en la pantalla, ofrece a los desarrolladores el control total sobre estilos y formato del sistema, separando el contenido de la presentación.

Framework CSS: Intenta aliviar el exceso de carga asociado con actividades comunes al trabajar con hojas de estilos y su intención es principalmente ahorrar tiempo y minimizar el riesgo de errores en el desarrollo de aplicaciones.

Tema: Cada tema se compone de plantillas que definen la visualización específica de bloques, menú y páginas en general, aplicando estilos CSS.

Interfaz de usuario: Medio con que cuenta el usuario para comunicarse con una máquina, equipo u ordenador y comprender la información que se quiere visualizar.

Personalización: La posibilidad de adaptar un sistema a las exigencias y necesidades propias de los usuarios, los aspectos gráficos, de apariencia y aquellos relacionados con los aspectos organizativos y estructurales.

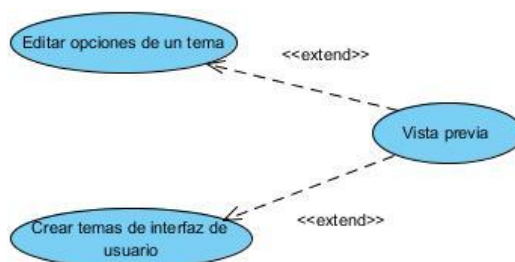
Widgets: Son pequeñas aplicaciones o programas. Entre sus objetivos está brindar fácil acceso a las aplicaciones frecuentemente usadas. Pueden interactuar con servicios e información distribuida en Internet; pueden ser relojes en pantalla, notas, calculadoras, calendarios, agendas y juegos, entre otros.



Anexos

En este apartado se expondrán los artefactos generados durante el avance de la investigación que no fueron expuestos en el cuerpo del documento.

Anexo 1. Ejemplo del patrón de caso de uso Adición



Anexo 2. Descripción de los casos de uso

Anexo 2.1: Descripción del CU Listar temas de interfaz de usuario

CU Listar temas de interfaz de usuario.	
Objetivo	Listar temas de interfaz de usuario que se encuentren cargados en la plataforma.
Actores	Administrador central
Resumen	El caso de uso se inicia cuando el actor accede a la opción de Listar temas de interfaz de usuario. El sistema muestra los temas que se encuentran cargados en la plataforma. Permite consultar la estructura de un tema, así como realizar cualquier acción sobre él. En caso de acceder a eliminar un tema el sistema permitirá eliminar los que no cumplan con los requisitos del usuario. El caso de uso termina.
Complejidad	Media
Prioridad	Crítico
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado.



Postcondiciones	Se ha listado los temas de interfaz de usuario por el actor.	
Flujo de eventos		
Flujo básico: “Listar temas de interfaz de usuario”.		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona la opción de listar los temas de interfaz de usuario.	
2.		<p>Muestra un listado de los temas que se encuentran cargados en la plataforma con los datos:</p> <ul style="list-style-type: none"> • Título. • Descripción. • Dirección del tema. • Fecha de publicación. <p>Permite:</p> <ul style="list-style-type: none"> • Crear tema. <p>Y de cada tema permite:</p> <ul style="list-style-type: none"> • Editar • Eliminar • Ver vista previa • Posicionar Widgets
3	El actor selecciona la opción de Crear un tema.	
		<p>Brinda la posibilidad de crear un tema. <u>Ver CU “Crear tema de interfaz de usuario”.</u></p> <p>El caso de uso termina.</p>



Flujos alternos

*.a El actor selecciona la opción de editar un tema.

	Actor	Sistema
1		Brinda la posibilidad de editar un tema. <u>Ver CU "Editar opciones de un temas"</u> .
2		El caso de uso termina.

*.b El actor selecciona la opción de eliminar un tema.

	Actor	Sistema
1		Brinda la posibilidad de eliminar un tema. <u>Ver Sección 1: "Eliminar tema de interfaz de usuario"</u> .
2		El caso de uso termina

*.c El actor selecciona la opción de ver vista previa de un tema.

	Actor	Sistema
1		Brinda la posibilidad de visualizar un tema. <u>Ver CU "Vista previa"</u> .
2		El caso de uso termina.

*.d El actor selecciona la opción de Posicionar los Widgets de los temas.

	Actor	Sistema
1.		Brinda la posibilidad de posicionar los Widgets de los tema. <u>Ver CU "Posicionar Widgets"</u> .
2.		El caso de uso termina.



Sección 1: Eliminar temas de interfaz de usuario.

Flujo básico <Eliminar temas de interfaz de usuario>

	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona la opción eliminar un tema de la plataforma.	
2.		Muestra un mensaje de confirmación. Y permite: <ul style="list-style-type: none"> • Aceptar • Cancelar
3.	Selecciona la opción de Aceptar.	
4.		Regresa a la vista anterior, elimina el tema y muestra un mensaje de información.
5.		El caso de uso termina.

Flujos alternos

*.a El actor selecciona la opción de Cancelar.

	Actor	Sistema
1.		Regresa a la vista anterior.
2.		El caso de uso termina.

Anexo 2.2: Descripción del CU Editar opciones del tema

CU Editar opciones del tema



Objetivo	Editar las opciones de los temas de interfaz de usuario para el subsistema Bachiller.	
Actores	Administrador central	
Resumen	El caso de uso se inicia cuando el administrador central selecciona la opción que le permite editar un tema. El actor puede realizar cualquier modificación sobre el tema. Si el actor accede a modificar las opciones del tema, el sistema mostrará los elementos a modificar. Si el actor accede a visualizar la vista previa del tema, el sistema mostrará el tema con las modificaciones realizadas, y una vez realizados los cambios, guardará las modificaciones, terminando así el caso de uso.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	<p>Debe haberse generado el escritorio de trabajo del usuario autenticado.</p> <p>Se debe acceder a la página de administración de Materias.</p> <p>Para editar un tema debe haberse listado los temas existentes en la plataforma.</p>	
Postcondiciones	Se ha editado el tema de interfaz de usuario por el actor.	
Flujo de eventos		
Flujo básico: “Editar opciones del tema”.		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona la opción que le permite editar las opciones del tema.	
2.		Muestra las opciones del tema a modificar (General, Tipografías, Tablas, Formularios, Botones, Alertas, Barra de navegación,



		<p>Paginado, Layouts, Editor y Módulos)</p> <p>En caso de acceder a editar las opciones generales se podrá modificar:</p> <ul style="list-style-type: none">• Nombre del tema• Descripción• Imagen• Color del fondo• Color del texto• Color del capítulo• Imagen del banner• Imagen del fondo• Imagen de cerrar• Color de los vínculos• Color de los vínculos visitados <p>Y permite:</p> <ul style="list-style-type: none">• Guardar.• Guardar y cerrar.• Cancelar.• Ver vista previa.
3	Modifica las opciones generales y accede a guardar.	
4		Valida los datos.
5		Actualiza los cambios.



6		Muestra un mensaje de información. El caso de uso termina.
Flujos alternos		
*a El actor selecciona la opción de Guardar y cerrar.		
	Actor	Sistema
1		Actualiza los cambios.
2		Regresa a la vista anterior.
3.		El caso de uso termina.
*b El actor selecciona la opción de Cancelar.		
	Actor	Sistema
		Elimina los datos creados.
		Regresa a la vista anterior.
		El caso de uso termina.
*c El actor selecciona la opción de ver Vista Previa.		
	Actor	Sistema
1		Muestra la vista previa del tema. <u>Ver CU: "Vista Previa"</u> .
2		El caso de uso termina.
2.a El actor selecciona editar las Tipografías.		
	Actor	Sistema
1.		Muestra los campos a modificar: <ul style="list-style-type: none"> • Tamaño de la fuente h1



		<ul style="list-style-type: none"> • Tamaño de la fuente h2 • Tamaño de la fuente h3 • Tamaño de la fuente base • Familia de las fuentes
2.		El caso de uso termina.
2.b El actor selecciona editar las Tablas.		
	Actor	Sistema
1.		Muestra los campos a modificar: <ul style="list-style-type: none"> • Color de fondo • Color del acento del fondo • Color del fondo flotante • Color del borde
2.		El caso de uso termina.
2.c El actor selecciona editar los Formularios.		
	Actor	Sistema
1.		Muestra los campos a modificar: <ul style="list-style-type: none"> • Color de fondo • Color de borde • Radio del borde
2.		El caso de uso termina.
2.d El actor selecciona editar los Botones.		
	Actor	Sistema
1.		Muestra los campos a editar de los Botones primarios, de éxito y error:



		<ul style="list-style-type: none"> • Color de fondo • Color de fondo visitado
2.		El caso de uso termina.
2.e El actor selecciona editar las Alertas.		
	Actor	Sistema
1.		<p>Muestra los campos a editar de las alertas de advertencia, de error, de éxito y de información:</p> <ul style="list-style-type: none"> • Color del fondo • Color del texto • Color del borde
2.		El caso de uso termina.
2.f El actor selecciona editar la Barra de navegación.		
	Actor	Sistema
1.		<p>Muestra los campos a editar de la Barra de navegación:</p> <ul style="list-style-type: none"> • Barra de navegación • Color del fondo • Color de fondo visitado • Color del borde • Color del vínculo • Color del vínculo visitado • Color del fondo activo • Color de los vínculos activos • Menú desplegable



		<ul style="list-style-type: none"> • Color del fondo • Color del borde • Color del vínculo • Color del fondo visitado
2.		El caso de uso termina.
2.g El actor selecciona editar el Paginado.		
	Actor	Sistema
1.		<p>Muestra los campos a editar del Paginado:</p> <ul style="list-style-type: none"> • Imagen de regreso • Imagen de regreso visitada • Imagen de inicio • Imagen de inicio visitada • Imagen de siguiente • Imagen de siguiente visitada • Imagen de fin • Imagen de fin visitada
2.		El caso de uso termina.
2.h El actor selecciona editar los Layouts.		
	Actor	Sistema
1.		<p>Muestra los campos a editar de los layouts donde se puede modificar el código php de los layouts que contienen los temas:</p> <ul style="list-style-type: none"> • Contenido • Columna derecha de Galería • Dos columnas de Galería



		<ul style="list-style-type: none"> • Aprendizaje • Menú izquierdo • Menú derecho • Emergente • Principal • Sin menú • Dos nuevos menú
2.		El caso de uso termina.
2.i El actor selecciona editar las hojas de estilo.		
	Actor	Sistema
1.		<p>Muestra los campos a editar de las Hojas de estilo:</p> <ul style="list-style-type: none"> • Principal • Módulos
2.		El caso de uso termina.
2.j El actor selecciona editar los Módulos.		
	Actor	Sistema
1.		<p>En caso de acceder a editar las opciones de los módulos se podrá modificar:</p> <p style="text-align: center;">Contenidos</p> <ul style="list-style-type: none"> • Imagen para texto aumentado • Imagen para texto disminuido • Imagen para texto original • Editar CSS <p style="text-align: center;">Tareas</p>



		<ul style="list-style-type: none"> • Imagen para Reactivo • Imagen para Softareas y Evidencias • Imagen para Recorridos Dirigidos • Editar CSS <p>Prácticas</p> <ul style="list-style-type: none"> • Editar CSS <p>Biblioteca</p> <ul style="list-style-type: none"> • Imagen para Galería • Imagen para Consultas • Imagen para Glosario e Historia • Imagen para Información de Interés • Editar hojas de estilo <p>Docente</p> <ul style="list-style-type: none"> • Imagen para Programas • Imagen para Orientaciones Metodológicas • Imagen para Administración del Aprendizaje • Imagen para Aportaciones del docente • Imagen para Reportes • Imagen para Comunicaciones • Editar CSS
2.		El caso de uso termina.
2.k Visualizar las imágenes que presentan las opciones a editar.		
	Actor	Sistema



1.		Muestra al lado de cada imagen dentro de las opciones a editar un vínculo que permite visualizar la imagen que contiene dicho elemento.
2.		El caso de uso termina.
4. a Existen datos incompletos.		
	Actor	Sistema
1		Muestra un mensaje de información.
2		Muestra un indicador sobre los campos vacíos. Regresa al paso 4 del flujo básico.
4. b Existen datos incorrectos.		
	Actor	Sistema
1		Muestra un mensaje de información.
2		Muestra un indicador sobre los campos incorrectos. Regresa al paso 4 del flujo básico.

Anexo 2.3: Descripción del CU Crear temas de interfaz de usuario

CU Crear temas de interfaz de usuario.	
Objetivo	Crear los temas de interfaz de usuario para el subsistema Bachiller.
Actores	Administrador central
Resumen	El caso de uso se inicia cuando el administrador central selecciona la opción que le permite crear un tema. El sistema carga el tema base para que el actor pueda realizar cualquier modificación sobre él. Si el actor accede a editar las opciones



	del tema, el sistema mostrará los elementos a modificar, guardará las modificaciones sin editar el tema base, solo la copia cargada, terminando así el caso de uso.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	<p>Debe haberse generado el escritorio de trabajo del usuario autenticado.</p> <p>Se debe acceder a la página de administración de Materias.</p> <p>Para crear un tema debe haberse listado los temas existentes en la plataforma.</p>	
Postcondiciones	Se ha creado el tema de interfaz de usuario por el actor.	
Flujo de eventos		
Flujo básico: “Crear temas de interfaz de usuario”.		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona la opción que le permite crear un tema.	
2.		<p>Realiza una copia al tema base mostrando las opciones del tema a modificar (General, Tipografías, Tablas, Formularios, Botones, Alertas, Barra de navegación, Paginado, Layouts, Editor y Módulos)</p> <p>En caso de acceder a editar las opciones generales se podrá modificar:</p> <ul style="list-style-type: none"> • Nombre del tema • Descripción



		<ul style="list-style-type: none"> • Imagen • Color del fondo • Color del texto • Color del capítulo • Imagen del banner • Imagen del fondo • Imagen de cerrar • Color de los vínculos • Color de los vínculos visitados <p>Y permite:</p> <ul style="list-style-type: none"> • Guardar. • Guardar y cerrar. • Cancelar. • Ver vista previa.
3	Modifica las opciones generales y accede a guardar.	
4		Valida los datos.
5		Actualiza los cambios.
6		Muestra un mensaje de información. El caso de uso termina.
Flujos alternos		
*a El actor selecciona la opción de Guardar y cerrar.		



	Actor	Sistema
1		Actualiza los cambios.
2		Regresa a la vista anterior.
3.		El caso de uso termina.
*b El actor selecciona la opción de Cancelar.		
	Actor	Sistema
		Elimina los datos creados.
		Regresa a la vista anterior.
		El caso de uso termina.
*c El actor selecciona la opción de ver Vista Previa.		
	Actor	Sistema
1		Muestra la vista previa del tema. <u>Ver CU: "Vista Previa"</u> .
2		El caso de uso termina.
2.a El actor selecciona editar las Tipografías.		
	Actor	Sistema
1.		Muestra los campos a modificar: <ul style="list-style-type: none"> • Tamaño de la fuente h1 • Tamaño de la fuente h2 • Tamaño de la fuente h3 • Tamaño de la fuente base • Familia de las fuentes



2.		El caso de uso termina.
2.b El actor selecciona editar las Tablas.		
	Actor	Sistema
1.		Muestra los campos a modificar: <ul style="list-style-type: none"> • Color de fondo • Color del acento del fondo • Color del fondo flotante • Color del borde
2.		El caso de uso termina.
2.c El actor selecciona editar los Formularios.		
	Actor	Sistema
1.		Muestra los campos a modificar: <ul style="list-style-type: none"> • Color de fondo • Color de borde • Radio del borde
2.		El caso de uso termina.
2.d El actor selecciona editar los Botones.		
	Actor	Sistema
1.		Muestra los campos a editar de los botones primarios, de éxito y error: <ul style="list-style-type: none"> • Color de fondo • Color de fondo visitado
2.		El caso de uso termina.
2.e El actor selecciona editar las Alertas.		



	Actor	Sistema
1.		<p>Muestra los campos a editar de las alertas de advertencia, de error, de éxito y de información:</p> <ul style="list-style-type: none"> • Color del fondo • Color del texto • Color del borde
2.		El caso de uso termina.
2.f El actor selecciona editar la Barra de navegación.		
	Actor	Sistema
1.		<p>Muestra los campos a editar de la barra de navegación:</p> <ul style="list-style-type: none"> • Barra de navegación • Color del fondo • Color de fondo visitado • Color del borde • Color del vínculo • Color del vínculo visitado • Color del fondo activo • Color de los vínculos activos • Menú desplegable • Color del fondo • Color del borde • Color del vínculo • Color del fondo visitado



2.		El caso de uso termina.
2.g El actor selecciona editar el Paginado.		
	Actor	Sistema
1.		<p>Muestra los campos a editar del paginado:</p> <ul style="list-style-type: none"> • Imagen de regreso • Imagen de regreso visitada • Imagen de inicio • Imagen de inicio visitada • Imagen de siguiente • Imagen de siguiente visitada • Imagen de fin • Imagen de fin visitada
2.		El caso de uso termina.
2.h El actor selecciona editar los Layouts.		
	Actor	Sistema
1.		<p>Muestra los campos a editar de los layouts donde se puede modificar el código php de los layouts que contienen los temas:</p> <ul style="list-style-type: none"> • Contenido • Columna derecha de Galería • Dos columnas de Galería • Aprendizaje • Menú izquierdo • Menú derecho • Emergente



		<ul style="list-style-type: none"> • Principal • Sin menú • Dos nuevos menú
2.		El caso de uso termina.
2.i El actor selecciona editar las hojas de estilo.		
	Actor	Sistema
1.		<p>Muestra los campos a editar de las hojas de estilo:</p> <ul style="list-style-type: none"> • Principal • Módulos
2.		El caso de uso termina.
2.j El actor selecciona editar los Módulos.		
	Actor	Sistema
1.		<p>En caso de acceder a editar las opciones de los módulos se podrá modificar:</p> <p>Contenidos</p> <ul style="list-style-type: none"> • Imagen para texto aumentado • Imagen para texto disminuido • Imagen para texto original • Editar el CSS <p>Tareas</p> <ul style="list-style-type: none"> • Imagen para Reactivo • Imagen para Softareas y Evidencias • Imagen para Recorridos Dirigidos • Editar CSS <p>Prácticas</p>



		<ul style="list-style-type: none"> • Editar CSS <p>Biblioteca</p> <ul style="list-style-type: none"> • Imagen para Galería • Imagen para Consultas • Interés • Editar hojas de estilo <p>Docente</p> <ul style="list-style-type: none"> • Imagen para Programas • Imagen para Glosario e Historia • Imagen para Información de • Imagen para Orientaciones Metodológicas • Imagen para Administración del Aprendizaje • Imagen para Aportaciones del docente • Imagen para Reportes • Imagen para Comunicaciones • Editar CSS
2.		El caso de uso termina.
2.k Visualizar las imágenes que presentan las opciones a editar.		
	Actor	Sistema
1.		Muestra al lado de cada imagen dentro de las opciones a editar un vínculo que permite visualizar la imagen que contiene dicho elemento.
2.		El caso de uso termina.



4. a Existen datos incompletos.

	Actor	Sistema
1		Muestra un mensaje de información.
2		Muestra un indicador sobre los campos vacíos. Regresa al paso 4 del flujo básico.

4. b Existen datos incorrectos.

	Actor	Sistema
1		Muestra un mensaje de información.
2		Muestra un indicador sobre los campos incorrectos. Regresa al paso 4 del flujo básico.



Anexo 2.4: Descripción del CU Vista previa

CU Vista previa.		
Objetivo	Ver vista previa de un tema.	
Actores	Administrador central	
Resumen	El caso de uso se inicia cuando el administrador central selecciona la opción que le permite ver la vista previa de un tema seleccionado. El sistema dará la posibilidad de mostrar el tema con las modificaciones realizadas y permitirá cerrar la vista actual terminando así el caso de uso.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado. Se debe acceder a la página de administración de Materias. Para ver la vista previa de un tema debe haberse listado los temas existentes en la plataforma.	
Postcondiciones	Se ha visualizado el tema seleccionado por el actor.	
Flujo de eventos		
Flujo básico: "Vista previa".		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona la opción de ver una Vista previa del tema.	
2.		Muestra en una nueva página el tema con los cambios realizados hasta el momento, mostrando la página inicial de Bachiller con la



		<p>barra de navegación, menú desplegable, fondo y banner. Brinda además la posibilidad de acceder a la visualización de las modificaciones de:</p> <ul style="list-style-type: none"> • Tipografía • Tablas • Formularios • Botones • Alertas • Paginado <p>Y permite:</p> <ul style="list-style-type: none"> • Cerrar
3	El actor visualiza los cambios y accede a Cerrar la vista previa.	
4.		Cierra la página de vista previa.
5.		El caso de uso termina.
Flujos alternos		
*a El actor selecciona ver las modificaciones de la Tipografía.		
	Actor	Sistema
1.		<p>Muestra el diseño que presentan los tipos de tipografía dentro del tema:</p> <ul style="list-style-type: none"> • Fuente h1 • Fuente h2 • Fuente h3 • Texto base • Estilo de los vínculos



2.		El caso de uso termina.
*b El actor selecciona ver las modificaciones de las Tablas.		
	Actor	Sistema
1.		Muestra el diseño que presentan las tablas dentro del tema, mostrando las filas, columnas y texto que la contienen.
2.		El caso de uso termina.
*c El actor selecciona ver las modificaciones de los Formularios.		
	Actor	Sistema
1.		Muestra el diseño que presentan los formularios dentro del tema: <ul style="list-style-type: none"> • Campo de entrada • Campo de selección • Área de texto
2.		El caso de uso termina.
*d El actor selecciona ver las modificaciones de los Botones.		
	Actor	Sistema
1.		Muestra el diseño que presentan los botones dentro del tema: <ul style="list-style-type: none"> • Botón primario • Botón de éxito • Botón de advertencia
2.		El caso de uso termina.
*e El actor selecciona ver las modificaciones de las Alertas.		



	Actor	Sistema
1.		Muestra el diseño que presentan las alertas dentro del tema: <ul style="list-style-type: none"> • Alertas de advertencia • Alertas de error • Alertas de éxito • Alertas de información
2.		El caso de uso termina.

*.f El actor selecciona ver las modificaciones del Paginado.

	Actor	Sistema
1.		Muestra el diseño que presentan las flechas del paginado dentro del tema: <ul style="list-style-type: none"> • Flecha anterior • Flecha siguiente • Flecha inicio • Flecha fin
2.		El caso de uso termina.

Anexo 2.5: Descripción del CU Posicionar Widgets

CU Posicionar Widgets.	
Objetivo	Posicionar los Widgets que presentan los temas de interfaz de usuario.
Actores	Administrador central
Resumen	El caso de uso se inicia cuando el administrador central selecciona la opción que le permite establecer la posición que contendrán los Widgets, dentro de los temas de interfaz de usuario. Si el actor accede a posicionar el sistema mostrará



	los Widgets existentes en la plataforma y dará la posibilidad de realizar los cambios dentro de sus campos, y una vez realizados los cambios, guardará las modificaciones, terminando así el caso de uso.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	<p>Debe haberse generado el escritorio de trabajo del usuario autenticado.</p> <p>Debe haberse listado previamente los temas que se encuentran en la plataforma.</p>	
Postcondiciones	Se han posicionado los Widgets de los temas de interfaz de usuario por el actor.	
Flujo de eventos		
Flujo básico: "Posicionar Widgets"		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona la opción eliminar un tema de la plataforma.	
2.		<p>Muestra un listado con los Widgets existentes en los temas, brindando la posibilidad de modificar la posición mostrando las áreas en que se pueden ubicar:</p> <ul style="list-style-type: none"> • Área superior izquierda • Área superior derecha • Área inferior izquierda • Área inferior derecha • Área del pie de página izquierda

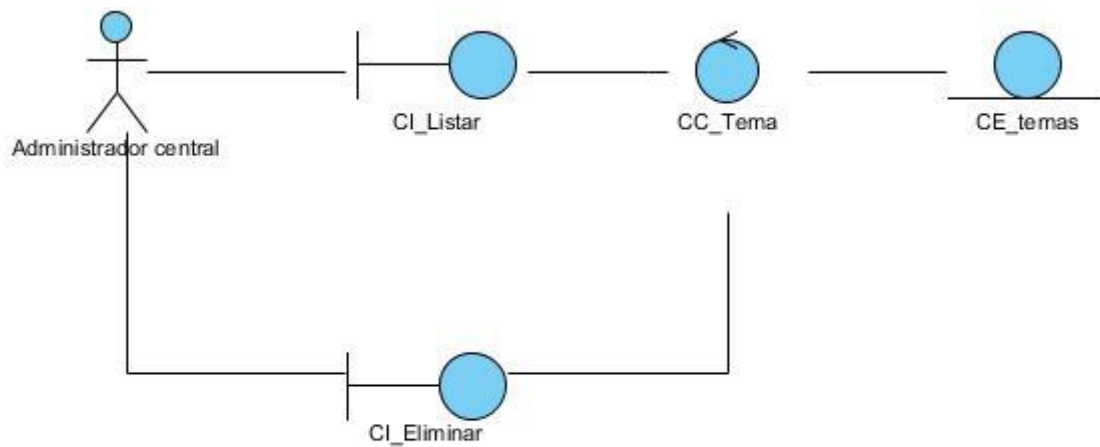


		<ul style="list-style-type: none"> • Área del pie de página derecha <p>Y permite:</p> <ul style="list-style-type: none"> • Cancelar la operación en cualquier momento. • Guardar actualizaciones. • Guardar y cerrar
3	Modifica la posición del Widgets y accede a guardar las actualizaciones.	
4		Muestra un mensaje de información.
5		Actualiza los cambios.
6		El caso de uso termina.
Flujos alternos		
*.a El actor selecciona la opción de Cancelar.		
	Actor	Sistema
1		Regresa a la vista anterior.
2		El caso de uso termina.
*.b El actor selecciona la opción de Guardar y cerrar.		
	Actor	Sistema
1.		Actualiza los cambios y regresa a la vista anterior.
2.		El caso de uso termina.

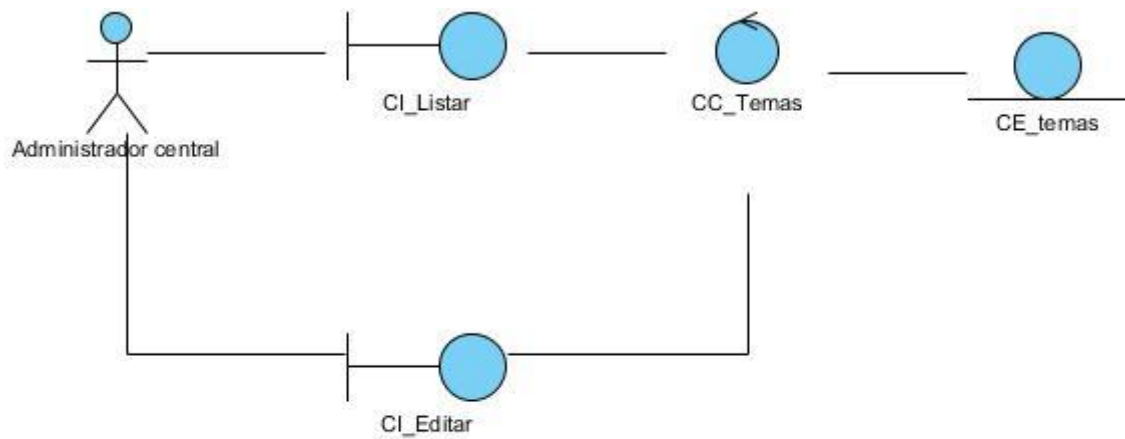


Anexo 3. Diagramas de Clases del Análisis

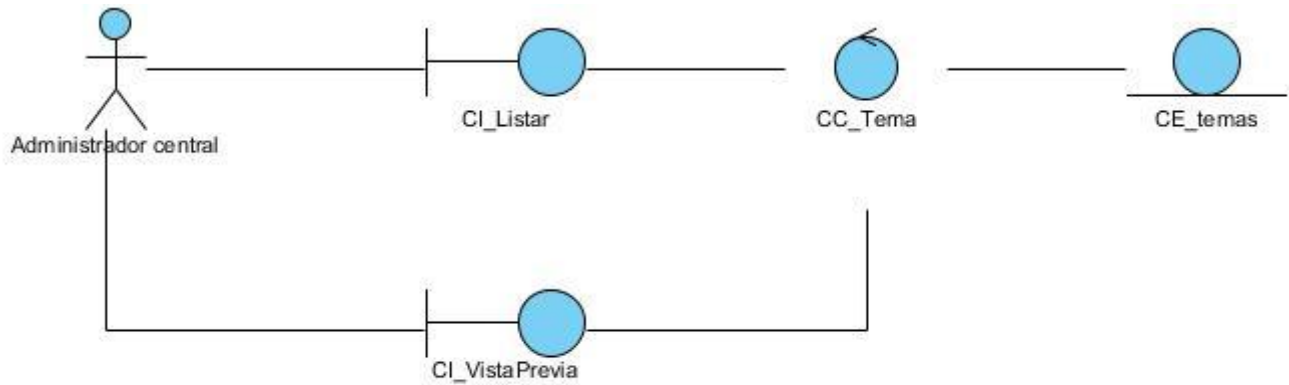
Anexo 3.1: DCA Listar temas de interfaz de usuario



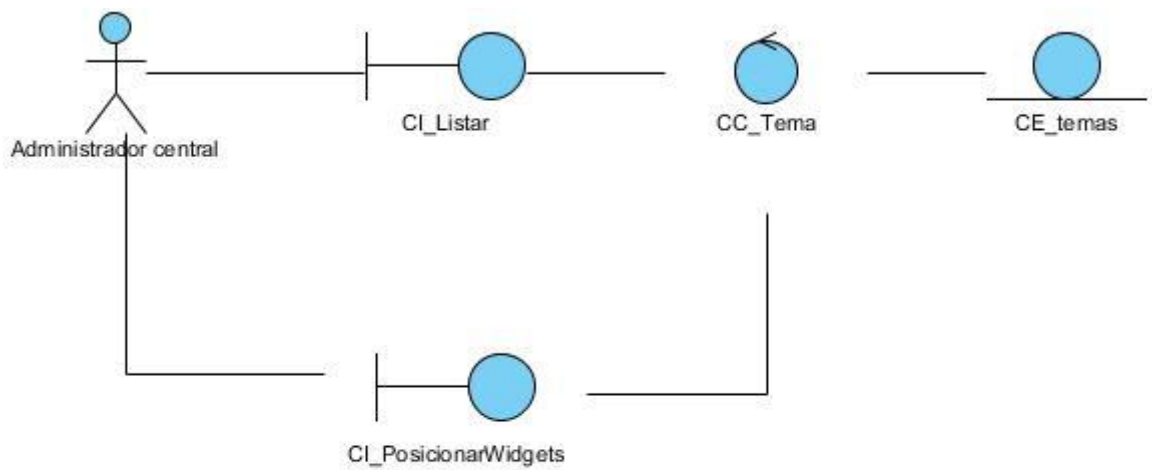
Anexo 3.2: DCA Editar opciones del tema



Anexo 3.3: DCA Vista previa



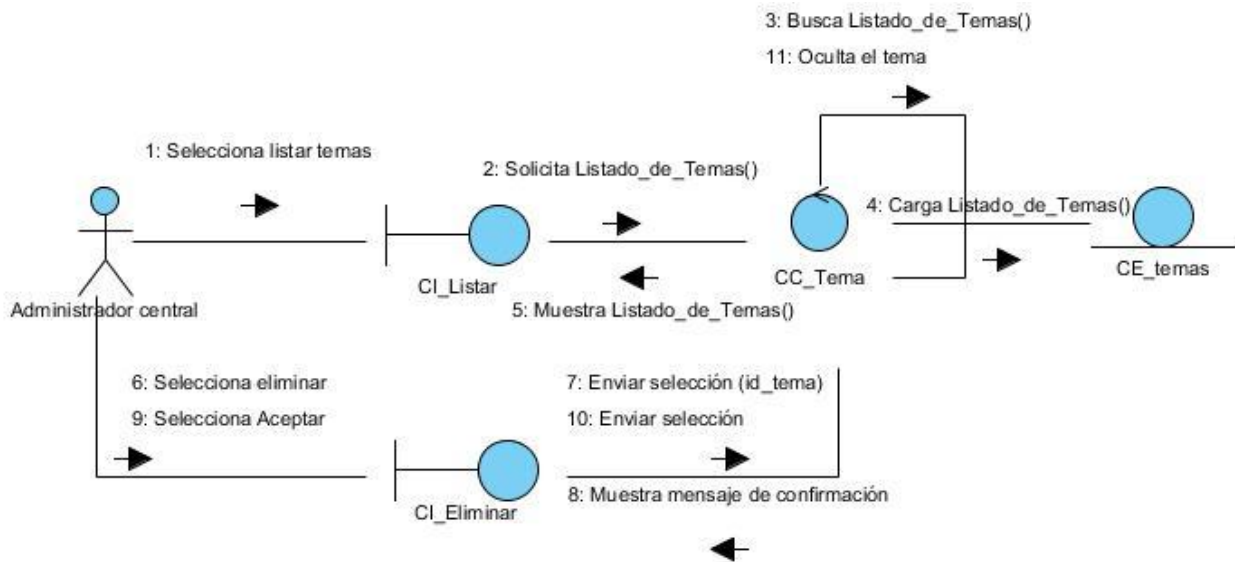
Anexo 3.4: DCA Posicionar Widgets



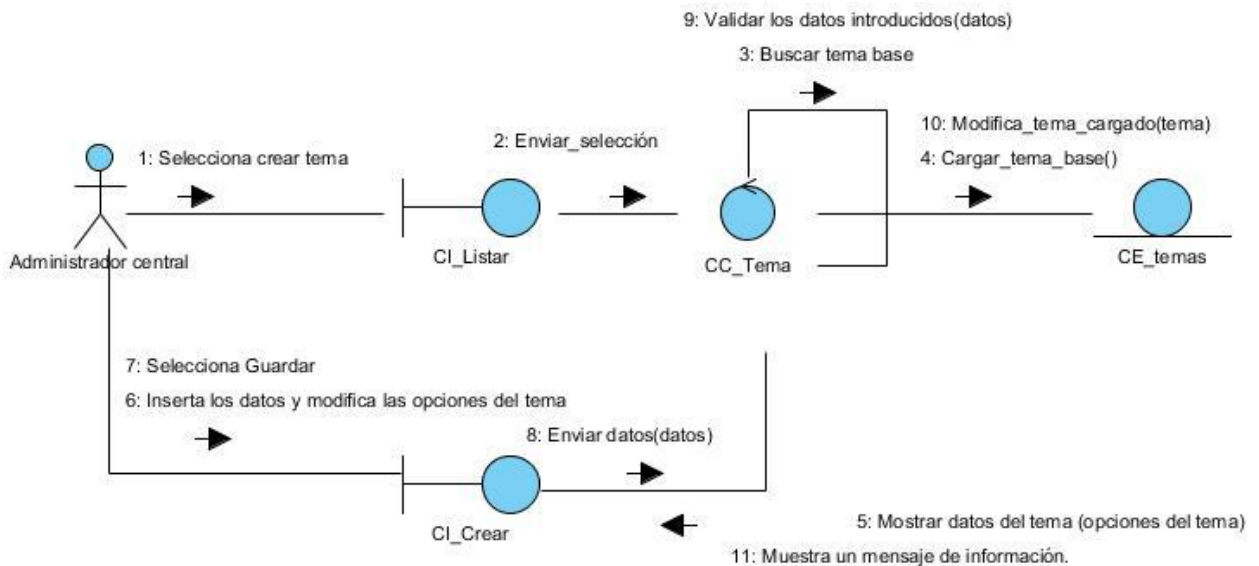


Anexo 4. Diagramas de Interacción del Análisis

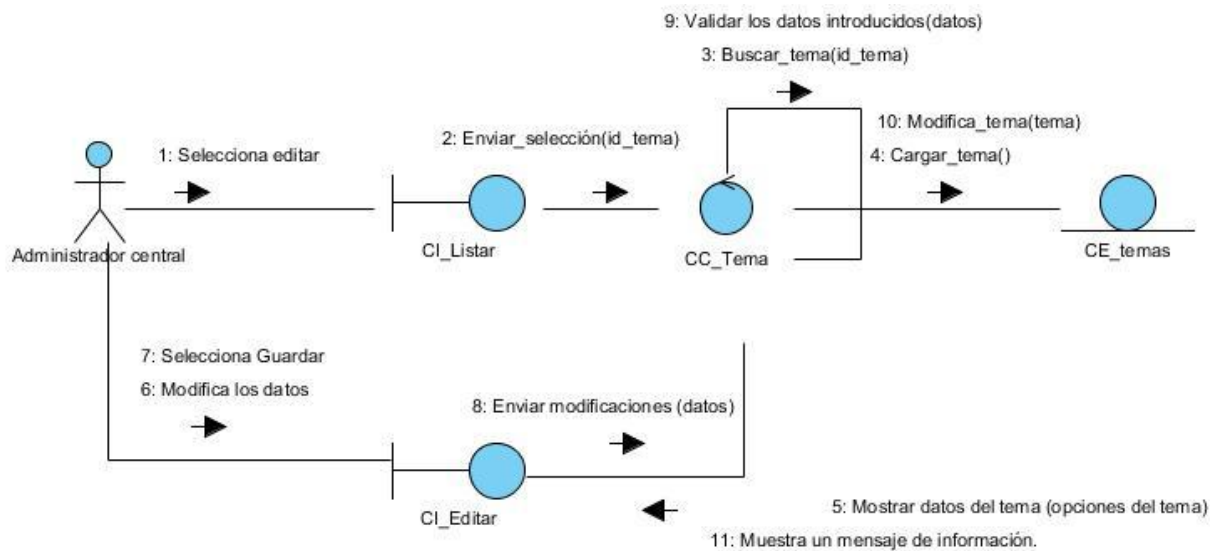
Anexo 4.1: DIA Listar temas de interfaz de usuario



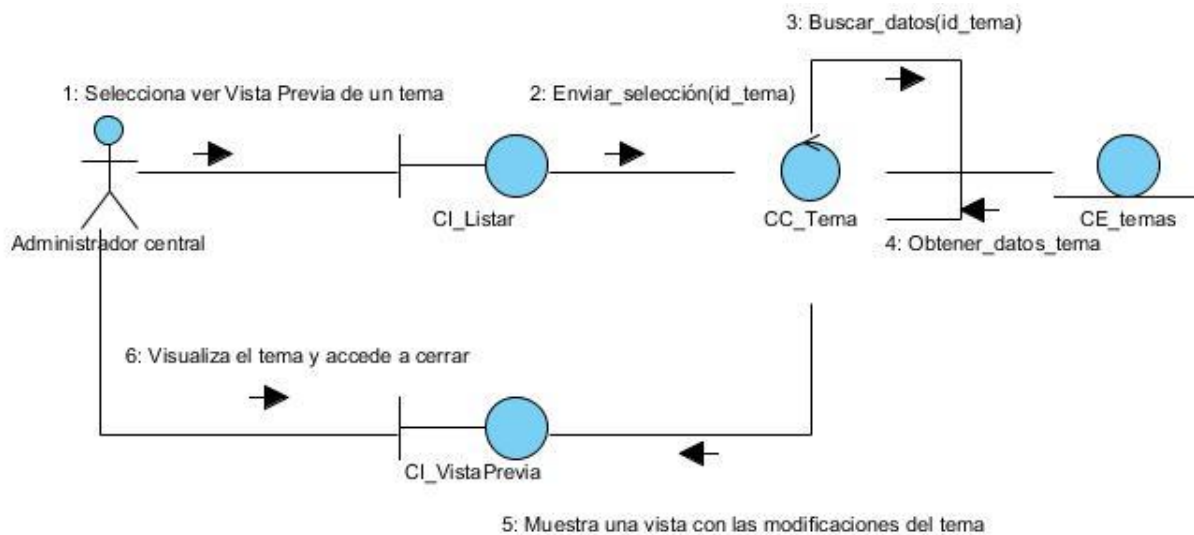
Anexo 4.2: DIA Crear temas de interfaz de usuario



Anexo 4.3: DIA Editar opciones del tema

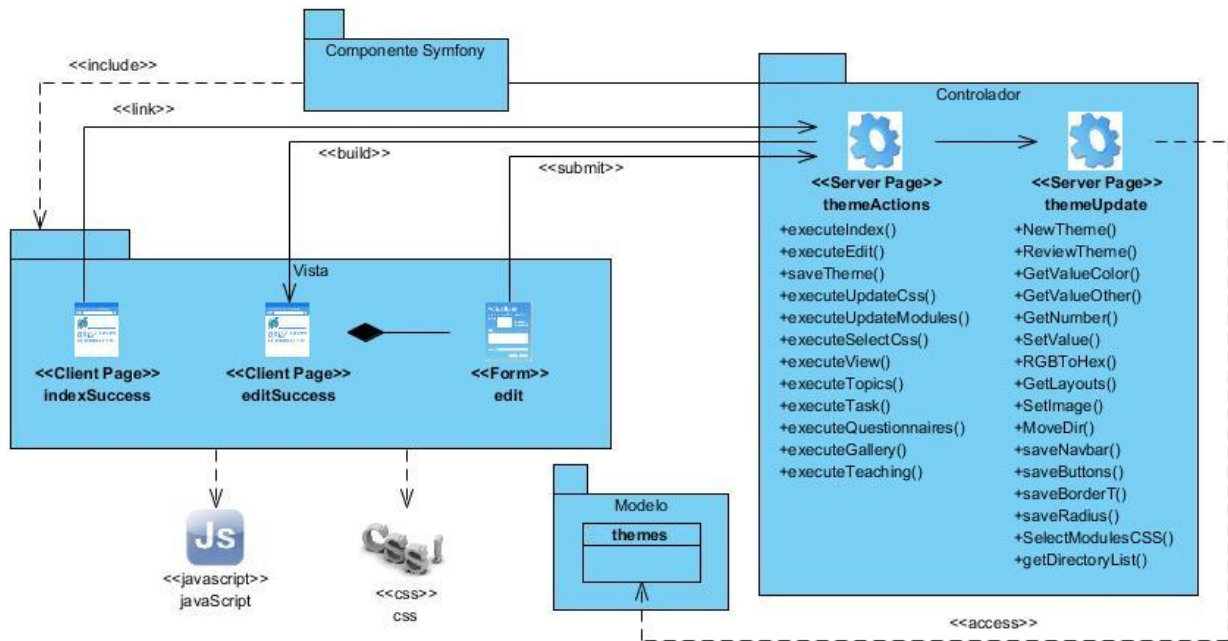


Anexo 4.4: DIA Vista previa

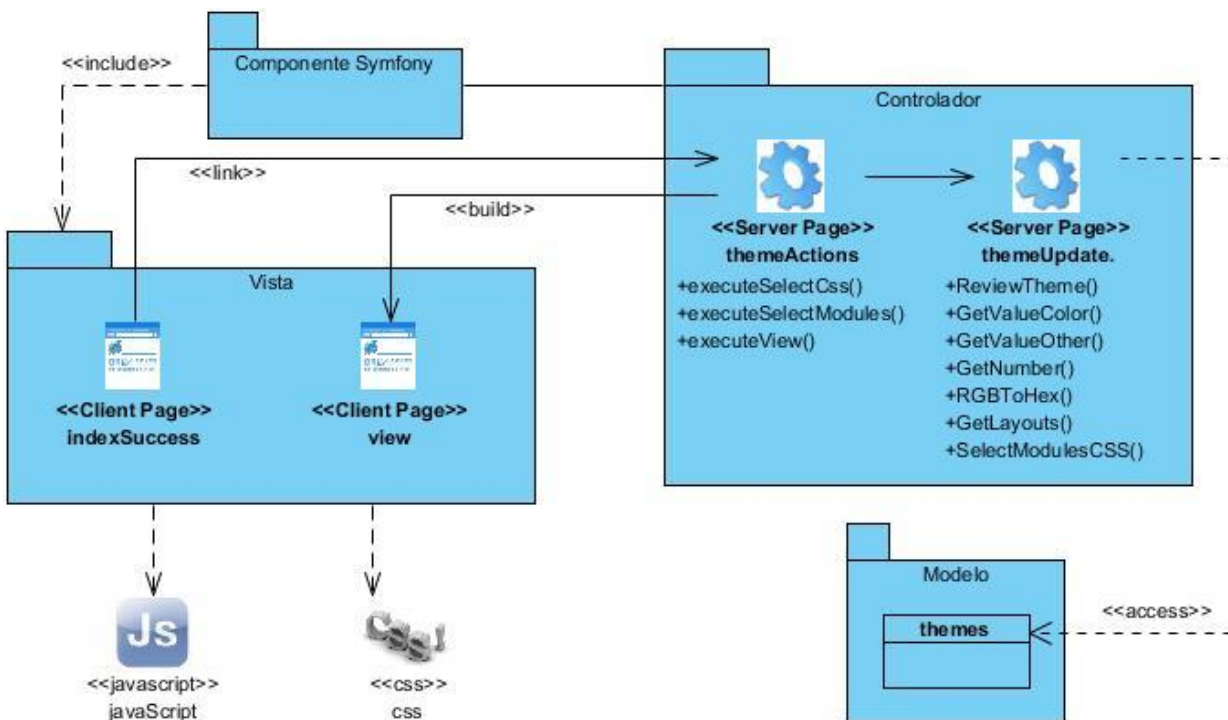




Anexo 5.2: DCD Editar opciones del tema

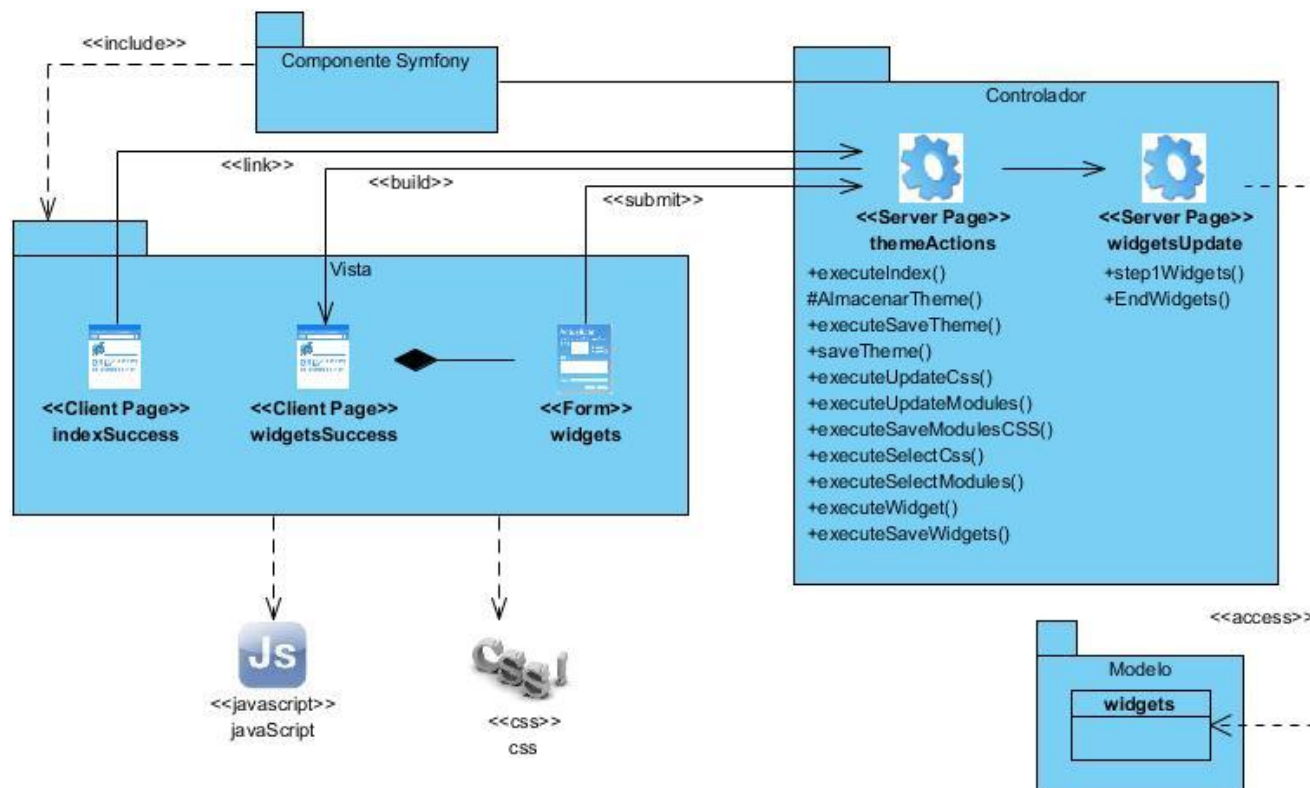


Anexo 5.3: DCD Vista previa





Anexo 5.4: DCD Posicionar Widgets



Anexo 6. Diseños de Casos de Prueba

Anexo 6.1: CP Crear temas de interfaz de usuario

Diseño de Casos de Prueba: CP Crear temas de interfaz de usuario.

Descripción General

El caso de uso se inicia cuando el administrador central selecciona la opción que le permite crear un tema. El sistema carga el tema base para que el actor pueda realizar cualquier modificación sobre él. Si el actor accede a editar las opciones del tema, el sistema mostrará los elementos a modificar, guardará las modificaciones sin editar el tema base, solo la copia cargada, terminando así el caso de uso.

Condiciones de Ejecución

Debe haberse generado el escritorio de trabajo del usuario autenticado.



Se debe acceder a la página de administración de Materias.

Para crear un tema debe haberse listado los temas existentes en la plataforma.

Escenario	Descripción	Nombre	Descripción	Imagen	Editar CSS	Tamaño de la fuente	Radio del borde	Respuesta del sistema	Flujo central
EC 1.1 Selecciona un tema para crear un tema.	El actor selecciona la opción que le permite crear un tema.	N/A	N/A	N/A	N/A	N/A	N/A	Realiza una copia al tema base mostrando las opciones del tema a modificar (General, Tipografías, Tablas, Formularios, Botones, Alertas, Barra de navegación, Paginado, Layouts, Editor y Módulos) En caso de acceder a editar las opciones generales se podrá modificar: <ul style="list-style-type: none"> • Nombre del tema • Descripción • Imagen • Color del fondo • Color del texto • Color del capítulo • Imagen del banner • Imagen del fondo • Imagen de cerrar • Color de los vínculos • Color de los vínculos visitados Y permite: <ul style="list-style-type: none"> • Guardar. • Guardar y cerrar. • Cancelar. • Ver vista previa. 	Módulo Materia/Temas/Crear nuevo tema
EC 1.2 Selecciona la opción de Guardar.	Modifica las opciones generales y accede a guardar.	V	V	V	V	V	V	Valida los datos. Actualiza los cambios. Muestra mensaje de información.	Módulo Materia/Temas/Crear nuevo tema/Guardar



EC 1.3 Selecciona la opción de Guardar y cerrar.	El actor accede a guardar y cerrar.	V	V	V	V	V	V	Actualiza los cambios. Regresa a la vista anterior.	Módulo Materia/Temas/Crear nuevo tema/Guardar y cerrar
EC 1.4 Selecciona la opción de Cancelar.	El actor selecciona la opción de Cancelar.							Elimina los datos creados. Regresa a la vista anterior.	Módulo Materia/Temas/Crear nuevo tema/Cancelar
EC 1.5 Selecciona la opción Ver vista previa.	El actor selecciona la opción de ver Vista Previa.							Muestra la vista previa del tema. <u>Ver CP: Vista previa.</u>	Módulo Materia/Temas/Crear nuevo tema/Vista previa
EC 1.6 Selecciona editar las tipografías.	El actor selecciona editar las tipografías.					V		Muestra los campos a modificar: <ul style="list-style-type: none"> • Tamaño de la fuente h1 • Tamaño de la fuente h2 • Tamaño de la fuente h3 • Tamaño de la fuente base • Familia de las fuentes 	Módulo Materia/Temas/Crear nuevo tema/Tipografía
EC 1.7 Selecciona editar las tablas.	El actor selecciona editar las tablas.							Muestra los campos a modificar: <ul style="list-style-type: none"> • Color de fondo • Color del acento del fondo • Color del fondo flotante • Color del borde 	Módulo Materia/Temas/Crear nuevo tema/Tablas
EC 1.8 Selecciona editar los formularios	El actor selecciona editar los formularios.						V	Muestra los campos a modificar: <ul style="list-style-type: none"> • Color de fondo • Color de borde • Radio del borde 	Módulo Materia/Temas/Crear nuevo tema/Formularios



EC 1.9 Selecciona editar los botones.	El actor selecciona editar los botones.						Muestra los campos a editar de los Botones primarios, de éxito y error: <ul style="list-style-type: none"> • Color de fondo • Color de fondo visitado 	Módulo Materia/Temas/Crear nuevo tema/Botones
EC 1.10 Selecciona editar las alertas.	El actor selecciona editar las alertas.						Muestra los campos a editar de las alertas de advertencia, de error, de éxito y de información: <ul style="list-style-type: none"> • Color del fondo • Color del texto • Color del borde 	Módulo Materia/Temas/Crear nuevo tema/Alertas
EC 1.11 Selecciona editar la barra de navegación.	El actor selecciona editar la barra de navegación.						Muestra los campos a editar de la Barra de navegación: <p>Barra de navegación</p> <ul style="list-style-type: none"> • Color del fondo • Color de fondo visitado • Color del borde • Color del vínculo • Color del vínculo visitado • Color del fondo activo • Color de los vínculos activos <p>Menú desplegable</p> <ul style="list-style-type: none"> • Color del fondo • Color del borde • Color del vínculo • Color del fondo visitado 	Módulo Materia/Temas/Crear nuevo tema/Barra de navegación
EC 1.12 Selecciona editar el paginado.	El actor selecciona editar el paginado.			V			Muestra los campos a editar del Paginado: <ul style="list-style-type: none"> • Imagen de regreso • Imagen de regreso visitada • Imagen de inicio • Imagen de inicio visitada • Imagen de siguiente • Imagen de siguiente visitada • Imagen de fin • Imagen de fin visitada 	Módulo Materia/Temas/Crear nuevo tema/Paginado



EC 1.13 Selecciona editar los layouts.	El actor selecciona editar los layouts.						<p>Muestra los campos a editar de los layouts donde se puede modificar el código php de los layouts que contienen los temas:</p> <ul style="list-style-type: none"> • Contenido • Columna derecha de Galería • Dos columnas de Galería • Aprendizaje • Menú izquierdo • Menú derecho • Emergente • Principal • Sin menú • Dos nuevos menú 	Módulo Materia/Temas/Crear nuevo tema/Layouts
EC 1.14 Selecciona editar las hojas de estilo	El actor selecciona editar las hojas de estilo.			V			<p>Muestra los campos a editar de las hojas de estilo:</p> <ul style="list-style-type: none"> • Principal • Módulos 	Módulo Materia/Temas/Crear nuevo tema/Editor
EC 1.15 Selecciona editar los módulos.	El actor selecciona editar los módulos.						<p>En caso de acceder a editar las opciones de los módulos se podrá modificar:</p> <p>Contenidos</p> <ul style="list-style-type: none"> • Imagen para texto aumentado • Imagen para texto disminuido • Imagen para texto original • Editar CSS <p>Tareas</p> <ul style="list-style-type: none"> • Imagen para Reactivo • Imagen para Softareas y Evidencias • Imagen para Recorridos Dirigidos • Editar CSS <p>Prácticas</p> <ul style="list-style-type: none"> • Editar CSS <p>Biblioteca</p> <ul style="list-style-type: none"> • Imagen para Galería • Imagen para Consultas • Imagen para Glosario e Historia • Imagen para Información de Interés • Editar hojas de estilo 	Módulo Materia/Temas/Crear nuevo tema/Módulos



								Docente • Imagen para Programas • Imagen para Orientaciones Metodológicas • Imagen para Administración del Aprendizaje • Imagen para Aportaciones del docente • Imagen para Reportes para Comunicaciones • Editar CSS	
EC 1.16 Selecciona visualizar imágenes actuales.	El actor selecciona visualizar las imágenes que presentan los elementos a editar.							Muestra al lado de cada imagen dentro de las opciones a editar un vínculo que permite visualizar la imagen que contiene dicho elemento.	Módulo Materia/Temas/Crear nuevo tema/Opciones/Ícono de ver imagen
EC 1.16 Datos incompletos	Existen datos incompletos.	I	I	I	I	I	I	Muestra un mensaje de información. Muestra un indicador sobre los campos vacíos.	Módulo Materia/Temas/Crear nuevo tema/Guardar ó Módulo Materia/Temas/Crear nuevo tema/Guardar y cerrar
		I	V	V	V	V	V		
		V	I	V	V	V	V		
		V	V	I	V	V	V		
EC 1.17 Datos incorrectos	Existen datos incorrectos	I	I	I	I	I	I	Muestra un indicador sobre los campos incorrectos.	Módulo Materia/Temas/Crear nuevo tema/Guardar ó Módulo Materia/Temas/Crear nuevo tema/Guardar y cerrar
		I	V	V	V	V	V		
		V	I	V	V	V	V		
		V	V	I	V	V	V		
		V	V	V	I	V	V		
		V	V	V	V	I	V		
		V	V	V	V	V	I		
		V	V	V	V	V	I		



Anexo 6.2: CP Editar opciones del tema

Diseño de Casos de Prueba: CP Editar opciones del tema.

Descripción General

El caso de uso se inicia cuando el administrador central selecciona la opción que le permite editar un tema. El actor puede realizar cualquier modificación sobre el tema. Si el actor accede a modificar las opciones del tema, el sistema mostrará los elementos a modificar. Si el actor accede a visualizar la vista previa del tema, el sistema mostrará el tema con las modificaciones realizadas, y una vez realizados los cambios, guardará las modificaciones, terminando así el caso de uso.

Condiciones de Ejecución

Debe haberse generado el escritorio de trabajo del usuario autenticado.

Se debe acceder a la página de administración de Materias.

Para editar un tema debe haberse listado los temas existentes en la plataforma.

Escenario	Descripción	Nombre	Descripción	Imagen	Editar CSS	Tamaño de la fuente	Radio del borde	Respuesta del sistema	Flujo central
EC 1.1 Selecciona editar un tema.	El actor selecciona la opción que le permite editar las opciones de un tema.	N/A	N/A	N/A	N/A	N/A	N/A	Muestra las opciones del tema a modificar (General, Tipografías, Tablas, Formularios, Botones, Alertas, Barra de navegación, Paginado, Layouts, Editor y Módulos) En caso de acceder a editar las opciones generales se podrá modificar: <ul style="list-style-type: none"> • Nombre del tema 	Módulo Materia/Temas/Editar



								<ul style="list-style-type: none"> • Descripción • Imagen • Color del fondo • Color del texto • Color del capítulo • Imagen del banner • Imagen del fondo • Imagen de cerrar • Color de los vínculos • Color de los vínculos visitados <p>Y permite:</p> <ul style="list-style-type: none"> • Guardar. • Guardar y cerrar. • Cancelar. • Ver vista previa. 	
EC 1.2 Selecciona la opción de guardar.	Modifica las opciones generales y accede a guardar.	V	V	V	V	V	V	<p>Valida los datos.</p> <p>Actualiza los cambios.</p> <p>Muestra mensaje de información.</p>	Módulo Materia/Temas/Editar/Guardar
EC 1.3 Selecciona la opción de guardar y cerrar.	El actor accede a guardar y cerrar.	V	V	V	V	V	V	<p>Actualiza los cambios.</p> <p>Regresa a la vista anterior.</p>	Módulo Materia/ Temas / Editar/ Guardar y cerrar
EC 1.4 Selecciona la opción de cancelar.	El actor selecciona la opción de cancelar.							<p>Elimina los datos creados.</p> <p>Regresa a la vista anterior.</p>	Módulo Materia/Temas/Editar/ Cancelar



EC 1.5 Selecciona la opción Ver vista previa.	El actor selecciona la opción de ver Vista Previa.						Muestra la vista previa del tema. <u>Ver CP: Vista previa.</u>	Módulo Materia/Temas/Editar/ Vista previa
EC 1.6 Selecciona editar las tipografías.	El actor selecciona editar las tipografías.					V	Muestra los campos a modificar: •Tamaño de la fuente h1 •Tamaño de la fuente h2 •Tamaño de la fuente h3 •Tamaño de la fuente base •Familia de las fuentes	Módulo Materia/Temas/Editar/ Tipografía
EC 1.7 Selecciona editar las tablas.	El actor selecciona editar las tablas.						Muestra los campos a modificar: • Color de fondo • Color del acento del fondo • Color del fondo flotante • Color del borde	Módulo Materia/Temas/Editar/Tablas
EC 1.8 Selecciona editar los formularios	El actor selecciona editar los formularios.					V	Muestra los campos a modificar: • Color del texto titular • Color de fondo • Color de borde • Radio del borde	Módulo Materia/ Temas/ Editar/Formularios
EC 1.9 Selecciona editar los botones.	El actor selecciona editar los botones.						Muestra los campos a editar de los Botones primarios, de éxito y error: • Color de fondo • Color de fondo visitado	Módulo Materia/Temas/Editar/ Botones
EC 1.10 Selecciona editar las alertas.	El actor selecciona editar las alertas.						Muestra los campos a editar de las alertas de advertencia, de error, de éxito y de información: • Color del fondo	Módulo Materia/Temas/Editar/Alertas



							<ul style="list-style-type: none"> • Color del texto • Color del borde 	
EC 1.11 Selecciona editar la barra de navegación.	El actor selecciona editar la barra de navegación.						<p>Muestra los campos a editar de la Barra de navegación:</p> <p>Barra de navegación</p> <ul style="list-style-type: none"> • Color del fondo • Color de fondo visitado • Color del borde • Color del vínculo • Color del vínculo visitado • Color del fondo activo • Color de los vínculos activos <p>Menú desplegable</p> <ul style="list-style-type: none"> • Color del fondo • Color del borde • Color del vínculo • Color del fondo visitado 	Módulo Materia/Temas/Editar/Barra de navegación
EC 1.12 Selecciona editar el paginado.	El actor selecciona editar el paginado.			V			<p>Muestra los campos a editar del Paginado:</p> <ul style="list-style-type: none"> • Imagen de regreso • Imagen de regreso visitada • Imagen de inicio • Imagen de inicio visitada • Imagen de siguiente • Imagen de siguiente visitada • Imagen de fin • Imagen de fin visitada 	Módulo Materia/Temas/Editar/ Paginado



EC 1.13 Selecciona editar los layouts.	El actor selecciona editar los layouts.							Muestra los campos a editar de los layouts donde se puede modificar el código php de los layouts que contienen los temas: <ul style="list-style-type: none"> • Contenido • Columna derecha de Galería • Dos columnas de Galería • Aprendizaje • Menú izquierdo • Menú derecho • Emergente • Principal • Sin menú • Dos nuevos menús. 	Módulo Materia/Temas/Editar/Layouts
EC 1.14 Selecciona editar las hojas de estilo	El actor selecciona editar las hojas de estilo.				V			Muestra los campos a editar de las Hojas de estilo: <ul style="list-style-type: none"> • Principal • Módulos 	Módulo Materia/Temas/Editar/Editor
EC 1.15 Selecciona editar los módulos.	El actor selecciona editar los módulos.							En caso de acceder a editar las opciones de los módulos se podrá modificar: <p>Contenidos</p> <ul style="list-style-type: none"> • Imagen para texto aumentado • Imagen para texto disminuido • Imagen para texto original • Editar CSS <p>Tareas</p> <ul style="list-style-type: none"> • Imagen para Reactivo • Imagen para Softareas y Evidencias • Imagen para Recorridos Dirigidos • Editar CSS <p>Prácticas</p>	Módulo Materia/Temas/Editar/Módulos



								<ul style="list-style-type: none"> • Editar CSS Biblioteca • Imagen para Galería • Imagen para Consultas • Imagen para Glosario e Historia • Imagen para Información de Interés • Editar hojas de estilo Docente • Imagen para Programas • Imagen para Orientaciones Metodológicas • Imagen para Administración del Aprendizaje • Imagen para Aportaciones del docente • Imagen para Reportes • Imagen para Comunicaciones • Editar CSS 	
EC 1.16 Selecciona visualizar imágenes actuales.	El actor selecciona visualizar las imágenes que presentan los elementos a editar.							Muestra al lado de cada imagen dentro de las opciones a editar un vínculo que permite visualizar la imagen que contiene dicho elemento.	Módulo Materia/Temas/Editar tema/Opciones/Ícono de ver imagen
EC 1.16 Datos incompletos	Existen datos incompletos.	I	I	I	I	I	I	Muestra un mensaje de información.	Módulo Materia/Temas/Editar/ Guardar
		I	V	V	V	V	V	Muestra un indicador sobre los campos vacíos.	ó Módulo Materia/Temas/Editar/ Guardar y cerrar
		V	V	I	V	V	V		



		V	V	V	I	V	V		
		V	V	V	V	I	V		
		V	V	V	V	V	I		
EC 1.17 Datos incorrectos	Existen datos incorrectos	I	I	I	I	I	I	Muestra un indicador sobre los campos incorrectos.	Módulo Materia/Temas/Editar/ Guardar ó Módulo Materia/Temas/Editar/ Guardar y cerrar
		I	V	V	V	V	V		
		V	I	V	V	V	V		
		V	V	I	V	V	V		
		V	V	V	I	V	V		
		V	V	V	V	I	V		
		V	V	V	V	V	I		

Anexo 6.3: CP Vista previa

Diseño de Casos de Prueba: CP Vista previa.			
Descripción General			
<p>El caso de uso se inicia cuando el administrador central selecciona la opción que le permite ver la vista previa de un tema seleccionado. El sistema dará la posibilidad de mostrar el tema con las modificaciones realizadas y permitirá cerrar la vista actual terminando así el caso de uso.</p>			
Condiciones de Ejecución			
<p>Debe haberse generado el escritorio de trabajo del usuario autenticado.</p> <p>Se debe acceder a la página de administración de Materias.</p> <p>Para ver la vista previa de un tema debe haberse listado los temas existentes en la plataforma.</p>			
Escenario	Descripción	Respuesta del sistema	Flujo central



EC 1.1 Selecciona visualizar un tema.	El actor selecciona la opción de ver una Vista previa del tema.	Muestra en una nueva página el tema con los cambios realizados hasta el momento, mostrando la página inicial de Bachiller con la barra de navegación, menú desplegable, fondo y banner. Brinda además la posibilidad de acceder a la visualización de las modificaciones de: <ul style="list-style-type: none"> • Tipografía • Tablas • Formularios • Botones • Alertas • Paginado Y permite: <ul style="list-style-type: none"> • Cerrar 	Módulo Materia/Temas/Vista Previa
EC 1.2 Selecciona la opción de Cerrar.	El actor visualiza los cambios y accede a Cerrar la vista previa.	Cierra la página de vista previa.	Módulo Materia/Temas/Vista Previa/ X
EC 1.3 Selecciona ver tipografía.	El actor selecciona ver las modificaciones de la tipografía.	Muestra el diseño que presentan los tipos de tipografía dentro del tema: <ul style="list-style-type: none"> • Fuente h1 • Fuente h2 • Fuente h3 • Texto base • Estilo de los vínculos 	Módulo Materia/Temas/Vista Previa/ Ejemplo: Tipografía
EC 1.4 Selecciona ver las tablas.	El actor selecciona ver las modificaciones de las tablas.	Muestra el diseño que presentan las tablas dentro del tema, mostrando las filas, columnas y texto que la contienen.	Módulo Materia/Temas/Vista Previa/ Ejemplo: Tablas
EC 1.5 Selecciona ver los formularios.	El actor selecciona ver las modificaciones de los formularios	Muestra el diseño que presentan los formularios dentro del tema: <ul style="list-style-type: none"> • Campo de entrada • Campo de selección • Área de texto 	Módulo Materia/Temas/Vista Previa/ Ejemplo: Formularios



EC 1.6 Selecciona ver los botones.	El actor selecciona ver las modificaciones de los botones.	Muestra el diseño que presentan los botones dentro del tema: <ul style="list-style-type: none"> • Botón primario • Botón de éxito • Botón de advertencia 	Módulo Previa/ Materia/Temas/Vista Ejemplo: Botones
EC 1.7 Selecciona ver las alertas.	El actor selecciona ver las modificaciones de las alertas.	Muestra el diseño que presentan las alertas dentro del tema: <ul style="list-style-type: none"> • Alertas de advertencia • Alertas de error • Alertas de éxito • Alertas de información 	Módulo Previa/ Materia/Temas/Vista Ejemplo: Alertas
EC 1.8 Selecciona ver el paginado.	El actor selecciona ver las modificaciones del paginado.	Muestra el diseño que presentan las flechas del paginado dentro del tema: <ul style="list-style-type: none"> • Flecha anterior • Flecha siguiente • Flecha inicio • Flecha fin 	Módulo Previa/ Materia/Temas/Vista Ejemplo: Paginado

Anexo 6.4: CP Posicionar Widgets

Diseño de Casos de Prueba: CP Posicionar Widgets.

Descripción General

El caso de uso se inicia cuando el administrador central selecciona la opción que le permite establecer la posición que contendrán los Widgets, dentro de los temas de interfaz de usuario. Si el actor accede a posicionar el sistema mostrará los Widgets existentes en la plataforma y dará la posibilidad de realizar los cambios dentro de sus campos, y una vez realizados los cambios, guardará las modificaciones, terminando así el caso de uso.

Condiciones de Ejecución

Debe haberse generado el escritorio de trabajo del usuario autenticado.

Debe haberse listado previamente los temas que se encuentran en la plataforma.

Escenario	Descripción	Respuesta del sistema	Flujo central



<p>EC 1.1 Selecciona Posicionar los Widgets.</p>	<p>El actor selecciona la opción de posicionar los Widgets.</p>	<p>Muestra un listado con los Widgets existentes en los temas, brindando la posibilidad de modificar la posición mostrando las áreas en que se pueden ubicar:</p> <ul style="list-style-type: none"> • Área superior izquierda • Área superior derecha • Área inferior izquierda • Área inferior derecha • Área del pie de página izquierda • Área del pie de página derecha <p>Y permite:</p> <ul style="list-style-type: none"> • Cancelar la operación en cualquier momento. • Guardar actualizaciones. • Guardar y cerrar 	<p>Módulo Materia/Temas/ Posicionar Widgets</p>
<p>EC 1.2 Selecciona la opción de Guardar.</p>	<p>Modifica la posición del Widgets y accede a guardar las actualizaciones.</p>	<p>Muestra un mensaje de información. Actualiza los cambios.</p>	<p>Módulo Materia/Temas/ Posicionar Widgets/ Guardar</p>
<p>EC 1.3 Selecciona la opción de Cancelar.</p>	<p>El actor selecciona la opción de Cancelar.</p>	<p>Muestra un mensaje de información. Regresa a la vista anterior.</p>	<p>Módulo Materia/Temas/ Posicionar Widgets/ Cancelar</p>
<p>EC 1.4 Selecciona la opción de Guardar y cerrar.</p>	<p>El actor selecciona la opción de Guardar y cerrar.</p>	<p>Actualiza los cambios y regresa a la vista anterior.</p>	<p>Módulo Materia/Temas/ Posicionar Widgets/ Guardar y cerrar</p>



Anexo 7. Documentos adicionales

Anexo 7.1: Datos de Contacto

Datos de Contacto

MSc. Yaillet Martínez Pérez (yaillet@uci.cu): Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2003. Pertenece al proyecto Alfaomega del Centro FORTES, profesora asistente de Práctica Profesional y MSc en Gestión de Proyecto Informáticos.

Ing. Yerandy Manso Guerra (ymguerra@uci.cu): Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2009. Pertenece al proyecto Alfaomega del Centro FORTES.