



Universidad de las Ciencias Informáticas

Facultad 4

**Componente para la creación de recursos educativos
empleando la arquitectura del marco de trabajo Xalix**

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

Autor: Luis Antonio Camilo Pérez

Tutores: Ing. Lizandra Hernández Hernández

Ing. Yunior Orosa Velázquez

La Habana, junio 2016

“Año 58 de la Revolución”

Declaro que soy el único autor del trabajo “Componente para la creación de recursos educativos empleando la arquitectura del marco del trabajo Xalix” y autorizo a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autor: Luis Antonio Camilo Pérez

Tutora: Ing. Lizandra Hernández Hernández

Tutor: Ing. Yunior Orosa Velázquez

“Desarrolla una pasión por aprender. Si lo haces, nunca dejarás de crecer”.

Anthony J. D'Angelo



Luis Antonio Camilo Pérez

A mis padres por su amor, confianza y apoyo incondicional, son lo que más quiero en la vida. A mis hermanos, abuelas, a mi abuelo que dios lo tenga en la gloria, amigos, amigas y personas que me ayudaron durante el proceso de formación tanto en la trayectoria estudiantil como en mi vida personal.

Luis Antonio Camilo Pérez

Es una satisfacción personal terminar algo que costó mucho al inicio, en este tiempo en la universidad descubrí que la vida es más bonita cuando se trabaja por las cosas que uno quiere junto a personas de buen corazón. Mi más profundo agradecimiento a mis padres por su amor, guía y apoyo incondicional en mis años de estudios. Estudiar en la universidad era uno de mis más grandes sueños y pude hacerlo realidad. Sus consejos me ayudaron a superar los momentos más difíciles en mi vida de estudiante y el trabajo que hicieron para que nunca me faltara nada es digno de reconocimiento, los amo. Ustedes siempre han sido para mí un gran ejemplo de constancia y por eso los quiero con todo mi corazón. Quiero darles las gracias por creer en mí y motivarme a ser un profesional, ahora soy graduado de la universidad y me siento listo para afrontar los retos de la vida. A mi esposa Liena por su cariño, apoyo y confianza con la cual he logrado culminar mi carrera profesional, por haber brindado su mano en las derrotas y logros de mi vida, haciendo de este triunfo más suyo que mío por la forma en la que guió mi vida con amor y energía. Gracias por tu ejemplo de superación incasable, por tu comprensión y confianza, por tu amor y amistad incondicional, porque sin su apoyo no hubiera sido posible esta realidad. A mis hermanos en especial a mi hermano Nolberto quien la ilusión de su vida ha sido verme convertido en un hombre de provecho. A mis abuelas Elsa y Bertha, a mi tíos y tías, en especial a mi tía Cari, a mis primos, a Teresa, Inés, Tirso y la familia de mi esposa que me han acogido como un hijo, gracias a su apoyo y consejo he llegado a realizar la más grande de mis metas. Por el cariño, comprensión y apoyo brindado en los momentos buenos y malos de mi vida, hago este triunfo compartido, sólo esperando que comprendan que mis ideales y esfuerzos son inspirados en cada uno de ustedes, gracias. Les agradezco sinceramente a mis tutores Lizandra y Yuniór por estar siempre dispuesto ayudarme en lo que me hiciera falta. A todos los estudiantes y profesores que compartieron momentos conmigo durante estos años, a todos mis compañeros de aula y en especial a Alexander, Reysel, Felo, Carlos Goncè, Carlos Galera, Nadia, Miguel, Cesar, Frank, Adrián, Marcos, Rosbel, Osvel, Daril, Taboada, a mis amistades que no se encuentran en la universidad Yuniór, Dariel, Olga, Vladi. De corazón Gracias.

Resumen

Esta investigación estudia el proceso de gestión de los recursos educativos, así como sus características, estándares, ventajas, desventajas y clasificación de los mismos. Los recursos educativos digitales en todas sus clasificaciones surgieron a partir de la necesidad de contar con contenidos que pudieran ser compartidos y reutilizados en diferentes contextos para apoyar el proceso de enseñanza-aprendizaje. La poca diversidad de recursos educativos creados a través de las herramientas desarrolladas en el Centro de Tecnologías para la Formación, evidenciaron la necesidad de crear e incorporar nuevos recursos desarrollados mediante la utilización de tecnologías y especificaciones definidas para el marco de trabajo Xalix. El objetivo de la presente investigación, es desarrollar un componente para la creación de recursos educativos de tipo: línea de tiempo, mapa conceptual y crucigrama empleando la arquitectura del marco de trabajo Xalix. Para su cumplimiento se realizó el análisis del estado del arte de los procedimientos para la creación de estos tres tipos de recursos educativos. La propuesta de solución se desarrolló guiada por la metodología AUP en su variante para la Universidad, basándose en el framework Symfony2 y PHP como lenguaje de desarrollo. Las pruebas realizadas permitieron comprobar y validar las funcionalidades del componente desarrollado, obteniéndose como resultado una solución que cumple con los objetivos planteados, dejando presente las ventajas que traen estos recursos educativos para las herramientas desarrolladas en el centro.

Palabras clave: crucigrama, framework, línea de tiempo, mapa conceptual, recursos educativos.

Índice

Introducción	1
Capítulo I. Fundamentación teórica	7
Introducción	7
1.1 Conceptos asociados al dominio del problema	7
1.1.1 E-learning.....	7
1.1.2 Generalidades de los recursos educativos.....	7
1.1.3 Características de los recursos educativos.....	8
1.1.4 Clasificación de los recursos educativos.....	9
1.1.5 Estándares <i>e-learning</i> de los recursos educativos.....	10
1.1.6 Ventajas y desventajas de los recursos educativos	11
1.2 Estudio de soluciones similares.....	12
1.2.1 FreeMind.....	13
1.2.2 Dipity	14
1.2.3 Hot Potatoes	15
1.3 Metodología, tecnologías y herramientas	17
1.3.1 Metodología de desarrollo de software	17
1.4 Herramientas y tecnologías	18
1.4.1 Herramienta CASE.....	18
1.4.2 Lenguajes para el desarrollo	19
1.4.3 Framework de desarrollo.....	22
1.4.4 Servidor web	24
1.4.5 Sistema Gestor de Base de Datos	25
1.4.6 Entorno Integrado de Desarrollo (IDE).....	26

1.5	Conclusiones parciales	26
Capítulo II. Propuesta de solución		27
	Introducción	27
2.1	Descripción del sistema	27
2.2	Modelo de dominio	27
2.3	Especificación de requisitos	28
2.3.1	Requisitos funcionales	29
2.3.2	Requisitos no funcionales	30
2.4	Historias de usuarios	31
2.5	Descripción de la arquitectura	32
2.6	Patrones de diseño	33
2.6.1	Diagrama de Clases de Diseño (DCD)	35
2.6.2	Diagramas de secuencia del diseño (DS)	36
2.7	Modelo de datos	39
2.8	Conclusiones parciales	40
Capítulo III. Implementación y pruebas.....		41
	Introducción	41
3.1	Modelo de implementación	41
3.1.1	Diagrama de componentes	41
3.1.2	Diagrama de despliegue	43
3.2	Estándares de codificación	44
3.3	Pruebas del software	46
3.3.1	Niveles de prueba	46
3.3.2	Métodos de prueba	47
3.3.3	Diseño de Casos de prueba.....	48

3.3.4 Resultados obtenidos.....	54
3.4 Conclusiones parciales.....	55
Conclusiones generales.....	57
Recomendaciones.....	58
Bibliografía.....	59

Índice de figuras

Figura 1. Modelo de dominio. 28

Figura 2. Patrón arquitectónico MVC..... 33

Figura 3. Diagrama de clase de diseño de la línea de tiempo. 36

Figura 4. DS_Seleccionar recurso de tipo línea de tiempo..... 37

Figura 5. DS_Añadir evento. 37

Figura 6. DS_Modificar evento. 38

Figura 7. DS_Ver eventos. 38

Figura 8. DS-Eliminar eventos..... 39

Figura 9. DS_Ver línea de tiempo..... 39

Figura 10. Modelo físico de la base de datos. 40

Figura 11. Diagrama de componente de la línea de tiempo. 42

Figura 12. Diagrama de despliegue..... 43

Figura 13. Gráfico de los resultados obtenidos..... 55

Índice de tablas

Tabla 1. HU_Seleccionar línea de tiempo.....	31
Tabla 2. HU_Gestionar eventos.....	32
Tabla 3. CP_Gestionar eventos.....	48
Tabla 4. Variables correspondientes al CP efectuado.	53
Tabla 5. Resultados obtenidos.....	55

Introducción

En la actualidad el desarrollo constante de las tecnologías de la información y las comunicaciones (TIC), ha impulsado el empleo de medios tecnológicos para realizar actividades educativas, con el fin de diversificar y hacer menos tradicional el proceso educativo. Las buenas prácticas educativas exigen acciones como diseñar y emplear nuevas estrategias que faciliten el aprendizaje, conocer las dificultades más comunes de los estudiantes y entender cómo las TIC pueden apoyar a superarlas. (1)

La educación se proyecta hoy con un nuevo paradigma en el proceso de enseñanza-aprendizaje a partir de la creación de herramientas digitales, permitiendo el perfeccionamiento de los escenarios educativos en la mayoría de las instituciones educacionales. El uso de estas ha potenciado un nuevo modo de aprendizaje llamado *e-learning* o aprendizaje electrónico, definido por los autores Foix ¹ y Zavando ² en *“aquella actividad que utiliza de manera integrada y pertinente computadoras y redes de comunicación, en la formación de un ambiente propicio para la construcción de la experiencia de aprendizaje”* (2). Con el surgimiento de este nuevo modo de aprendizaje surge la necesidad de crear y distribuir recursos educativos, encaminados a facilitar la puesta a disposición de una amplia oferta de contenidos digitales para todos los niveles de enseñanza (3). Al hablar de recursos educativos se hace referencia: *“(...) a todo medio material o conceptual que se utiliza como apoyatura en la enseñanza, con la finalidad de facilitar o estimular el aprendizaje”*. (4)

Los recursos educativos pueden ser clasificados como objetos o recursos digitales en ellos: *“el valor de la información es impredecible, depende de aspectos muy diversos, y, sobre todo, de quien la utilice...”*. (5) Su valor educativo radica en utilizarlos como un método integral de comunicación dentro de los planes de estudio, con la intención generar nuevos conocimientos a través de elementos de la didáctica y la pedagogía. Su poder transformador reside en la facilidad con que tales recursos pueden compartirse a través de internet. Estos pueden ser cursos completos o materiales de los mismos, multimedias, objetos de aprendizajes o cualquier elemento con fines educativos. (6) (7)

En Cuba, específicamente en la Universidad de las Ciencias Informáticas (UCI), se encuentra el Centro de Tecnologías para la Formación (FORTES), el cual tiene como misión desarrollar tecnologías que permitan ofrecer servicios y productos, para brindar soluciones de formación aplicando las TIC, a todo tipo de

¹**Cristian Alberto Foix Weishaupt:** Ingeniero Civil en Informática, con más de 9 años de experiencia en implantaciones y puesta

²**Sonia Zavando:** Directora Centro Desarrollo Educación Media en INACAP de Chile, Miembro Comité Técnico Fondef Educación y Ciencias Sociales en CONICYT.

instituciones con diferentes modelos de formación y condiciones tecnológicas (8). El centro formalizó un marco de trabajo denominado Xalix³, con el propósito de disminuir la diversidad tecnológica de las soluciones. Además, de lograr la integración y reutilización de los componentes de la herramienta de autor web CRODA, que se especializa en la creación de objetos de aprendizajes reutilizables, accesibles, duraderos e interoperables. La gestión de los hiperentornos de aprendizaje de la plataforma educativa ZERA y el repositorio de objetos de aprendizajes RHODA, donde se almacenan y gestionan los objetos de aprendizaje creados por los diferentes gestores existentes en la universidad.

En estudios anteriores realizados a la herramienta de autor web CRODA y la plataforma educativa ZERA se observó que permiten crear un aproximado de 19 tipos de recursos educativos (9) (10), agrupados en categorías de actividades de autoevaluación, multimedia, recursos interactivos y estructurales. A diferencia de las múltiples herramientas existentes en el ámbito internacional que apoyan el proceso de enseñanza-aprendizaje, dentro de las más utilizadas se encuentran los sistemas de gestión de aprendizaje Moodle⁴ y Claroline⁵, y las herramientas de autor *Hot-Potatoes*⁶ y *Exe-Learning*⁷. El catálogo de recursos y actividades ofrecidos por estas es aproximadamente de 50, siendo una cifra considerable con respecto a CRODA y ZERA (11) (12) (13) (14). La poca diversidad de recursos educativos creados a través de las herramientas desarrolladas en el centro evidenciaron la necesidad de crear e incorporar nuevos recursos desarrollados mediante tecnologías libres, permitiendo aprovechar el potencial de las TIC en relación con la evolución de los recursos educativos. Teniendo como finalidad establecer herramientas con una oferta suficiente y variada de materiales formativos que despierten interés a nivel educacional, empresarial e institucional, además de impulsar la madurez y el desarrollo de software en la Universidad.

A partir de la situación planteada se identifica como **problema a resolver**: ¿Cómo contribuir con la creación de recursos educativos para el Centro de Tecnologías para la Formación? Por lo que se define como **objetivo general** desarrollar un componente para la creación de recursos educativos de tipo: línea

³ **Xalix** es un marco de trabajo creado en el Centro de Tecnologías para la Formación de la Universidad de las Ciencias Informáticas que cuenta con un conjunto de componentes orientados a garantizar la seguridad, instalación, migración y guardar la configuración de un sistema de software.

⁴ **Moodle** es un software informático para crear y gestionar plataformas educativas.

⁵ **Claroline** es una plataforma de aprendizaje y trabajo virtual de software libre que permite a los formadores construir cursos en línea y gestionar las actividades de aprendizaje y colaboración en la Web.

⁶ **Hot Potatoes** es una herramienta de autor para la generación de ejercicios y materiales interactivas multimedia.

⁷ **EXeLearning** es una herramienta de autor gratuita y de código abierto que permite la generación de contenidos Web con fines educativos.

de tiempo, mapa conceptual y crucigrama empleando la arquitectura del marco de trabajo Xalix. Este problema enmarca como **objeto de estudio** el proceso de gestión de recursos educativos. Delimitando como **campo de acción** la creación de recursos educativos.

El objetivo general se desglosa los siguientes **objetivos específicos**:

- ❖ Elaborar el marco teórico conceptual relacionado con el proceso de gestión de recursos educativos.
- ❖ Realizar el diseño del componente para la creación de recursos educativos.
- ❖ Implementar un componente para la creación de recursos educativos empleando la arquitectura del marco de trabajo Xalix.
- ❖ Realizar pruebas para garantizar la funcionalidad del componente.

Para guiar el proceso de desarrollo se hizo necesario formular las siguientes preguntas científicas:

- ❖ ¿Cuáles son las características de los recursos educativos?
- ❖ ¿Cuáles son las tecnologías necesarias para el desarrollo del componente?
- ❖ ¿Cómo desarrollar un componente para creación los recursos educativos empleando la arquitectura del marco de trabajo Xalix?

En el desarrollo del proceso de investigación se tuvieron en cuenta varios **métodos tradicionales investigativos**, los cuales se mencionan a continuación quedando evidenciado de qué forma fueron utilizados.

Métodos teóricos (15):

- ❖ **Análisis histórico-lógico:** se utilizó para analizar la evolución histórica de las soluciones similares, las tendencias más recientes de las aplicaciones web y basándose en esos datos complementar las características necesarias o deseables en la solución propuesta.
- ❖ **Analítico-sintético:** para el estudio de las fuentes bibliográficas existentes referente a los recursos educativos, identificando los conceptos más importantes y necesarios para dar solución a la problemática planteada.
- ❖ **Modelación:** facilitó la representación explícita de la solución propuesta durante el proceso de desarrollo de recursos educativos.
- ❖ **Análisis documental:** se empleó para consultar literatura especializada en el objeto de estudio.

Métodos empíricos (15):

- ❖ **Método comparativo:** se estableció una comparación entre las herramientas digitales más reconocidas y el ambiente desarrollado.

Estructura capitular:

Capítulo I: Fundamentación teórica.

Se presentan marco teórico conceptual de la investigación realizada, así como el estudio de soluciones similares, métodos de investigativos, tecnologías, herramientas y metodologías utilizadas para el desarrollo de la solución propuesta.

Capítulo II: Propuesta de solución.

Se definen y describen los requisitos funcionales y no funcionales. Además, se elaboran los artefactos correspondientes al diseño de la propuesta de solución, como resultado de la realización del flujo de trabajo de la metodología seleccionada.

Capítulo III: Implementación y pruebas.

Se abordan los aspectos relacionados con la implementación del sistema y el proceso de pruebas utilizado. Se implementan todas las funcionalidades identificadas, para lograr un componente que permita cumplir el objetivo general planteado. Se detallan las pruebas que se realizaron al componente, con el propósito de darle cumplimiento de los requisitos funcionales definidos.

Capítulo I. Fundamentación teórica

Introducción

En este capítulo se definen los principales conceptos que serán abordados durante la investigación. Se destacan las principales características de los elementos asociados a la temática a investigar, así como el entorno internacional donde se desarrollan. Además, se describe la metodología de desarrollo de *software*, las herramientas y tecnologías que serán utilizadas en la propuesta de solución.

1.1 Conceptos asociados al dominio del problema

1.1.1 E-learning

Las TIC desempeñan un papel esencial en los tiempos modernos, un resultado directo de su aplicación lo constituye el surgimiento del *e-learning*, término que literalmente consiste en la vía de obtención de conocimientos por cualquier medio electrónico. Es una manera cómoda de proporcionarles a las personas materiales para su auto preparación tales como documentos, videos u otros materiales de interés educativo sin la necesidad de estar en presencia física de algún profesor. Es una de las opciones actuales más utilizada para atender la necesidad de aprendizaje permanente. (16)

Dadas sus características y el soporte tecnológico que lo respalda, el aprendizaje electrónico brinda la facilidad que el usuario pueda elegir sus horarios de estudio. Se ha convertido en una opción óptima para aquellas personas autónomas que trabajen y quieran estudiar en sus momentos libre, posibilitando no solo aprender conceptos nuevos sino también afianzar conocimientos y habilidades, aumentado así la autonomía y la motivación de los estudiantes por diferentes temas (17). La educación basada en el empleo del *e-learning*, soluciona preocupaciones tradicionales como proporcionar la comunicación adecuada entre los miembros de instituciones educativas. En este tipo de aprendizaje, las herramientas digitales juegan un papel fundamental en el trabajo colaborativo. Favorecen a perfeccionar el proceso de creación de recursos educativos que hoy constituyen una fuente nutritiva de conocimiento para los usuarios. (16)

1.1.2 Generalidades de los recursos educativos

Las tecnologías de la información y la comunicación han potenciado la creación, distribución y recepción de contenidos educativos, con el fin de generar nuevas formas de representación de la información. La digitalización de los materiales (imagen, sonido, textos verbales, infografía) procedentes de diferentes fuentes permite que puedan reutilizarse y combinarse en variadas herramientas digitales. El empleo de los recursos o materiales formativos en las estas herramientas se considera una guía para el aprendizaje,

organizando los contenidos a transmitir y permitiendo alcanzar el desarrollo de habilidades en los usuarios; se pueden considerar como un entorno colaborativo y de expresión para los usuarios. (18)

Los recursos educativos digitales incluyen (19):

- ❖ **Contenidos formativos:** pueden ser cursos completos, software educativo, módulos de contenido, recopilaciones y publicaciones.
- ❖ **Herramientas:** software para desarrollar, utilizar, reutilizar y entregar el contenido formativo, incluyendo búsqueda y organización del contenido, las herramientas de desarrollo de contenidos y las comunidades educativas en línea.
- ❖ **Recursos de implementación:** licencias de propiedad intelectual para promover la publicación abierta de materiales, diseño de principios de buenas prácticas y traducción de contenidos.

Todos estos elementos son utilizados por los recursos educativos para proporcionarles a los usuarios habilidades procedimentales y a mejorar la persona en actitudes o valores. Son empleados para: informar sobre un tema, ayudar en la adquisición de un conocimiento, reforzar un aprendizaje, remediar una situación desfavorable, favorecer el desarrollo de una determinada competencia y evaluar conocimientos.

Atendiendo a su importancia para el aprendizaje estos deben ser (20):

- ❖ **Prácticos:** Aportar información práctica y realista.
- ❖ **Contextualizados:** Observar el contexto socioeconómico, cultural y lingüístico de los usuarios.
- ❖ **Bien escritos:** Concisión, univocidad, eliminación de redundancias e imprecisiones.
- ❖ **Ejemplificativos:** Nutrirse de ejemplos, casos de estudio y escenarios auténticos y relevantes.

1.1.3 Características de los recursos educativos

Los recursos educativos pueden ser adaptados a distintas herramientas digitales, gracias a la existencia de características puntuales que deben tener, la presente investigación hace referencia algunas como (21):

- ❖ **Multimedia:** son los recursos que integran soportes o procedimientos que emplean sonido, imágenes o textos para difundir información, especialmente si es de forma interactiva.
- ❖ **Interactividad:** las estrategias y metodologías de aprendizaje que se aplican a los recursos educativos digitales deben tener en cuenta las posibilidades que ofrece el carácter interactivo e inmersivo de este medio para la consecución de los objetivos pedagógicos marcados.
- ❖ **Accesibilidad:** los recursos educativos digitales deben permitir el acceso a materiales instruccionales independientemente de su localización.

- ❖ **Flexibilidad:** los recursos educativos digitales deben ser controlados por los usuarios en función de los diversos objetivos que puedan alcanzarse a partir de la estructura y organización elemental de sus contenidos de aprendizaje.
- ❖ **Adaptabilidad:** para ajustar la instrucción a las necesidades individuales de los estudiantes.
Reusabilidad: para integrar los componentes instruccionales en una variedad de aplicaciones, sistemas y contextos.
- ❖ **Modularidad:** la estructura y funcionalidad de los recursos educativos digitales deben responder a un modelo de organización modular.
- ❖ **Portabilidad:** los recursos educativos digitales deben seguir sistemas estándar de empaquetado y transferencia para que sea posible compartirlos e integrarlos con garantías y plena funcionalidad en distintos sistemas y entornos de explotación.
- ❖ **Interoperabilidad** los recursos educativos digitales deberán acompañarse de índices, criterios de etiquetado y catalogación eficaces y de calidad.

1.1.4 Clasificación de los recursos educativos

La clasificación de los recursos educativos digitales determina la existencia de múltiples tipologías. En esta investigación se recogen algunas orientadas a perseguir los siguientes objetivos (22):

- ❖ Agilizar las condiciones de reutilización de la información.
- ❖ Ponderar las cualidades para efecto de valoración.
- ❖ Unificar criterios entre distintas dependencias.
- ❖ Aumentar las posibilidades de integración con los medios.
- ❖ Gestionar de manera más equilibrada las colecciones.

Para cumplir con esos objetivos, se presta atención a la función que desempeñan los recursos educativos a partir de su empleo y aplicación, de los que resultan las tres categorías siguientes (22):

- ❖ **Informativos:** integran los documentos que contengan datos de utilidad y con propósito educativo. Justifican desde una teoría hasta datos específicos y concretos pasando por conceptos clave.
- ❖ **Apoyo educativo:** es una gama que incluye información que favorece la incorporación de conocimientos, pero que no tiene implícitamente una labor de instrucción.
- ❖ **Tratamiento educativo:** son contenidos que median el acceso a la información para procurar un aprendizaje formativo o informativo con un lenguaje apropiado según el usuario al que va dirigido.

La búsqueda de criterios de clasificación centrados en la finalidad o utilidad didáctica de los recursos educativos comprende las siguientes categorías (23):

- ❖ **Material de Apoyo:** con una cobertura no determinada (total o parcial) del currículo o del programa de contenidos, depende de los objetivos a alcanzar en el desarrollo del acto didáctico, suele carecer de actividades de aprendizaje y autoevaluación, y la interactividad no es un aspecto prioritario.
- ❖ **Curso en línea:** material organizado en una plataforma "informático educativa" o en un sitio web, con cobertura total de programa de estudios y actividades y recursos de aprendizaje.
- ❖ **Paquetes didácticos:** conjunto de materiales en diversos formatos que se complementan en la consecución de objetivos de aprendizaje comunes, con cobertura total del programa de estudios.
- ❖ **Multimedios:** aquellos materiales que presentan una forma integrada de textos, gráficos, sonidos y animaciones en un medio digital. Sus ventajas vienen dadas por la naturaleza y potencialidad del soporte: fomento de la relación entre docente y discente; puesta a disposición de actividades de aprendizaje y de estrategias de construcción del conocimiento mediante la interactividad.
- ❖ **Objetos de aprendizaje:** porción más pequeña de instrucción o información que puede, por sí sola, tener significado para el estudiante. Su característica principal es que son reusables, interoperables y accesibles.
- ❖ **Libro electrónico:** versión electrónica del libro impreso, que incorpora funcionalidades nuevas, como vínculos internos y externos al contenido, opciones de impresión, elementos audiovisuales o buscadores.
- ❖ **Vídeo educativo:** es un material audiovisual con cierto grado de utilidad en los proceso de enseñanza y de aprendizaje.

1.1.5 Estándares *e-learning* de los recursos educativos

Los estándares *e-learning* son conocidos como especificaciones o modelos de referencia que permiten regular la creación de procesos en forma de patrón, formato o referencia. Los estándares tiene como protagonistas a los recursos educativos digitales, para los que se descubren cambios que afectan a su naturaleza y tipología; modelos que marcan los pasos y las formas de su desarrollo; oportunidades que modifican los roles y procesos de su difusión y distribución; o transformaciones que se exigen para su aprovechamiento. (23)

Genéricamente, *"la estandarización es la actividad que tiene por objeto establecer, ante problemas reales o potenciales, disposiciones destinadas a usos comunes y repetidos, con el fin de obtener un nivel de ordenamiento óptimo en un contexto dado, que puede ser tecnológico, político o económico"*. (23)

Los problemas clásicos de la informática educativa y que afectan a los recursos educativos se traduce en la baja posibilidad de reutilización/adaptación de contenidos o aplicaciones cuando cambia algún factor como, por ejemplo, la plataforma o el contexto educativo. Los estándares vienen a intentar salvar, entre otros, estos frecuentes obstáculos a la durabilidad, la actualización, la modificación, la combinación, la difusión y los diversos usos de unos recursos digitales que ya han superado exigencias y niveles de calidad relacionados con lo didáctico, lo narrativo y lo normativo, pero que deben además soportar un cambio tecnológico. (24)

Las cualidades de los recursos educativos digitales estandarizados son (25):

- ❖ **Interoperabilidad:** capacidad para intercambiar y mezclar contenido de múltiples fuentes para usarlo directamente en distintos sistemas.
- ❖ **Reusabilidad:** el contenido puede ser agrupado, desagrupado y reutilizado de forma rápida y sencilla; ensamblarse y utilizarse en un contexto distinto a aquél para el que fueron inicialmente diseñados.
- ❖ **Portabilidad:** el contenido puede ser utilizado en plataformas de explotación de diversos fabricantes.
- ❖ **Perdurabilidad:** cuando se producen cambios tecnológicos, los contenidos no demandan cambios relevantes en su diseño, ni una inversión significativa para seguir siendo reutilizables e interoperables.
- ❖ **Accesibilidad:** el contenido apropiado para un usuario permite a éste el acceso en el momento justo y en el dispositivo correcto.
- ❖ **Gestionabilidad / Trazabilidad:** el contenido puede trasladar información sobre sí mismo, sobre el usuario y sobre la interacción de ambos al sistema, que la recupera y la traza.

1.1.6 Ventajas y desventajas de los recursos educativos

Los recursos educativos digitales traen consigo una transformación didáctica en la educación donde se pone en práctica una metodología dinámica e innovadora, permitiendo de esta manera un mayor entusiasmo e interés por parte de los usuarios en las diferentes disciplinas y materias que se imparten. (26)

Entre las ventajas que pueden señalarse sobre la aplicación de los recursos educativos digitales se pueden mencionar (26):

- ❖ **La motivación:** los usuarios encuentran mayor atracción por la clase o por la materia que se imparte si se utilizan recursos atractivos o si la clase es dinámica. El uso de las TICs, le permite a los usuarios aprender jugando, quizás esta ventaja es la más importante, ya que el profesor puede

ser muy buen comunicador pero si no tiene la motivación del grupo será muy difícil que consiga sus objetivos trazados.

- ❖ **Interés:** generalmente hay algunas materias como matemáticas y ciencia que no son muy aceptadas por algunos usuarios al considerarlas difíciles; sin embargo, con el uso de las TICs un docente puede lograr que los estudiantes pierdan el miedo sobre dichas materias que no son de su agrado.
- ❖ **Interactividad:** el usuario puede interactuar, comunicar, intercambiar experiencias con otros compañeros del aula e inclusive con su mismo profesor.

Es importante tomar en cuenta que el hacer uso de las nuevas tecnologías puede generar también ciertas dificultades o desventajas que deben de tomarse en cuenta. Como por ejemplo (26):

- ❖ **El tiempo:** el no saber buscar la información que se necesita puede ocasionar cierta pérdida de tiempo por la innumerable información que se pueda encontrar en los diferentes canales o sitios.
- ❖ **Fiabilidad de la información:** gran cantidad de información que encontramos en la web no son fiables o están mal elaboradas y no pueden ser utilizados como sitios de consulta o de apoyo a una investigación.
- ❖ **Aislamiento:** el uso constante de los recursos educativos digitales contribuye a que los usuarios se aislen de otras formas de comunicación. Por tanto se debe educar y enseñar a los usuarios que tan importante es la utilización de los recursos educativos digitales anteponiendo la vida socio-afectiva con los que nos rodean.

Con el tiempo el término ha venido no sólo a referirse los recursos educativos sino también a la formación, al software de gestión, las herramientas de desarrollo, y las licencias para publicar recursos digitales. En la actualidad es importante disponer de un abanico de herramientas que nos permitan la creación, publicación y gestión de los materiales educativos en formato digital a utilizar en la educación a distancia mediada por las TIC. (27)

1.2 Estudio de soluciones similares

En el ámbito internacional existen múltiples herramientas digitales que permiten la creación de recursos educativos como cursos o lecciones que sirven de apoyo en el proceso de enseñanza-aprendizaje. Estas permiten controlar el aprendizaje, mediante cuestionarios y marcadores que evalúan el conocimiento que el estudiante vaya adquiriendo (19). A continuación se realiza un análisis de algunas herramientas que permiten la creación de recursos educativos de tipo: línea de tiempo, mapa conceptual y crucigrama, con el propósito de identificar las funcionalidades que servirán como guía para el desarrollo del componente.

1.2.1 FreeMind

FreeMind es una herramienta para la elaboración y manipulación de mapas conceptuales. Es decir, una herramienta para organizar y estructurar las ideas, los conceptos, su relación entre ellos y su evolución. Puede ser utilizada en cualquier área del ámbito educativo y como mecanismo o forma de plasmar tormentas de ideas. Está basado en Java⁸ y es *Open Source*⁹, bajo licencia GPL¹⁰. Es un software ligero, sencillo de instalar, configurar y utilizar. (28)

Las principales características de *FreeMind* son las siguientes (28):

- ❖ Permite la utilización de enlaces HTML en los nodos.
- ❖ Soporta el plegado y desplegado de los nodos.
- ❖ Arrastre/Copia inteligente, es decir, el usuario puede mover los nodos y su información y estilo asociado. También para múltiples nodos seleccionados, arrastrar textos, enlaces y archivos asociados.
- ❖ Permite publicar los mapas en Internet como páginas HTML o XHTML si se desea incluir la imagen.
- ❖ Tiene la opción de incluir íconos en el mapa conceptual para diferenciar ideas/conceptos específicas o relacionadas, además, permite incluir íconos simplemente como decoración.
- ❖ Utiliza formato XML lo que facilita exportar los mapas conceptuales a otras aplicaciones.
- ❖ Es multilinguaje, con lo cual soporta el español.

Los beneficios de crear mapas conceptuales con *FreeMind* son (28):

- ❖ Potenciar la destreza mental para obtener ideas brillantes.
- ❖ Solucionar problemas.
- ❖ Persuadir personas y negociar con ellas.
- ❖ Recordar cosas.
- ❖ Planificar objetivos personales
- ❖ Adquirir control sobre tu vida.

⁸ **Java**: es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems

⁹ **Open Source** (Código abierto) es el término con el que se conoce al software distribuido y desarrollado libremente.

¹⁰ **GPL**: Licencia Pública General de GNU, posibilita la modificación y redistribución del software, pero únicamente bajo esa misma licencia

- ❖ Organizar una presentación y escribir un informe.
- ❖ Aprobar exámenes.

Los conceptos básicos sobre los que se basa *FreeMind* son (28):

Utilización de un cuadro de texto que se utiliza para almacenar información denominado nodo.

- ❖ Los nodos están conectados mediante líneas curvas llamadas bordes.
- ❖ El primer nodo creado es el nodo raíz.
- ❖ Un mapa se construye añadiendo nodos al nodo raíz.
- ❖ Para editar un nodo hacer clic sobre él.

Nodos hijos y hermanos

- ❖ Un nodo hijo se coloca siempre a un nivel más bajo que el padre.
- ❖ Un nodo hermano se coloca al mismo nivel que su nodo de referencia.
- ❖ El nodo raíz sólo puede tener nodos hijos o secundarios.

Bordes

- ❖ Son las líneas que unen los nodos estableciendo las relaciones entre ellos. Pueden tener diferentes diseños y además se pueden personalizar.

1.2.2 Dipity

El *dipity* o *timeline*, es una herramienta de la Web 2.0 que sirve para crear líneas de tiempo de forma visual, de diferentes etapas de la historia. Con esta herramienta podemos crear líneas de tiempo multimedia, insertando texto, imágenes o vídeo.

Para fines educativos, es una herramienta que facilita la construcción de conocimientos utilizando diferentes tipos de lenguajes de expresión: textual, visual y auditivo. El *dipity*, puede ser de mucha utilidad no sólo como entretenimiento o de uso personal, sino también como estrategia pedagógica, sobre todo para la enseñanza de la historia. Igualmente, facilita la organización de información histórica y el aprendizaje colaborativo. (29)

Cuenta con fortalezas como (29):

- ❖ Permite cuatro modos de visionado: Línea de tiempo, álbum, lista de eventos y mapa; en éste genera un mapa personalizado de Google Maps con los datos y localizaciones de la línea.
- ❖ Se puede ver en modo pantalla completa.
- ❖ Las fechas pueden expresar desde los años hasta los segundos.

- ❖ Cada evento tiene su propia URL.
- ❖ Es la única que permite enlazar un evento con otro dentro de la misma línea o de una diferente.
- ❖ Genera automáticamente eventos a partir de fuentes externas (marcadores, YouTube, RSS).
- ❖ Genera un canal RSS, para ello sólo hay que añadir "/RSS" al final de la URL.
- ❖ El editor de texto acepta HTML básico y por lo tanto podemos insertar enlaces en el texto.
- ❖ Acepta una gran cantidad de elementos multimedia: imágenes, vídeo, audio y mapas (se insertan con gran facilidad).
- ❖ Admite la cooperación de varios editores.
- ❖ Funciones sociales: además de los comentarios de los usuarios, incluye listas de favoritos, notificaciones a amigos, blog y una fantástica sincronización con software social.
- ❖ Es la única que permite elegir el fondo en la cuenta gratuita.
- ❖ Visualmente es bastante clara y atractiva.
- ❖ Se inserta perfectamente en blogs, wikis y otros sitios web.

En términos de uso, una vez creada una cuenta en la herramienta ya podemos comenzar a confeccionar una línea de tiempo desde el botón "**Create a Timeline**". Completamos los siguientes datos: Título, descripción, categoría, zona horaria, imágenes en miniatura y permisos. Después podemos continuar haciendo clic en "**Continue and add events**". Para añadir un evento se selecciona la opción "**Add event**" luego completamos los datos de cada evento que se adicione. (29)

1.2.3 Hot Potatoes

Hot Potatoes es una herramienta de autor desarrollada por el Centro de Humanidades de la Universidad de Victoria (UVIC), en Canadá. Consta de varios programas o esquemas predeterminados. Es un recurso interesante para la generación de ejercicios y materiales interactivos multimedia. Es sencilla de utilizar y el uso de los recursos educativos sirve como un instrumento al servicio de las estrategias metodológicas, pues ayuda en la construcción del conocimiento. (14)

Esta herramienta de autor presenta características para su uso como (14):

- ❖ **Simplicidad:** crear o modificar ejercicios con *Hot Potatoes* es fácil e intuitivo, y está al alcance de todos: autodidactas o no, expertos en tecnología o reticentes a ella.
- ❖ **Aplicabilidad:** sirve para cualquier materia y nivel educativo.

Características relativas a su difusión:

- ❖ **Universalidad:** por la posibilidad que ofrece de traducir automáticamente la interfaz a varias lenguas, entre ellas algunas peninsulares. Podemos comprobar este hecho tanto en la página web

de *Hot Potatoes* y los tutoriales como dentro del mismo programa, los ejercicios generados son páginas html y se pueden difundir por Internet fácilmente, además, es de forma gratuita.

Permiten el diseño de los siguientes tipos de actividades:

- ❖ Preguntas tipo test de respuestas múltiples, cortas, híbridas y multiselección: programa *JQuiz*.
- ❖ Ejercicios de rellenar huecos: programa *JCloze*.
- ❖ Ejercicios tipo crucigramas para trabajar terminología específica: programa *JCross*.
- ❖ Ejercicios de emparejamiento u ordenación: programa *JMatch*.
- ❖ Ejercicios de ordenación de letras, palabras o fragmentos de texto: programa *JMix*.

Interacción con el ejercicio de tipo crucigrama o *JCross*:

Esta aplicación crea crucigramas a partir de palabras introducidas al configurar el ejercicio. Se puede cambiar el tamaño de la cuadrícula y se incluyen pistas y definiciones para la resolución del ejercicio.

Podemos elaborarlo de dos maneras diferentes:

Una primera manera es introducir manualmente las palabras en la posición que deseamos.

La otra opción es generar automáticamente la disposición de la cuadrícula debiendo indicar únicamente las palabras a incluir.

Añadimos las palabras deseadas y configuramos el tamaño de la cuadrícula. Una vez terminado pulsamos al botón "Crear la cuadrícula" para generar el ejercicio.

Una vez que hemos generado la cuadrícula con las palabras habrá que añadir las pistas que permitan la resolución del ejercicio. Para ello pulsaremos la opción "Añadir pistas" y editamos las pistas para cada término.

Análisis del estudio de soluciones similares

El análisis realizado permitió determinar las funcionalidades que serán utilizadas como base para el desarrollo de los componentes para la creación de recursos educativos.

Fueron tomadas como referencia las siguientes funcionalidades:

El empleo de cuadro de texto para almacenar información, la conexión mediante líneas, el soporte de plegado y desplegado de los nodos fueron tomados como ejemplo para el desarrollo del recurso de tipo mapa conceptual de la presente investigación, proporcionando mayor interactividad con el recurso, y brindando la posibilidad de establecer múltiples formas de representación.

Para la creación del recurso de tipo línea de tiempo fue tomada como referencia le forma de crear las líneas de tiempo y de mostrar la información de sus eventos, la incorporación de nuevos eventos a partir

de datos como (Título, Descripción, Categoría) además, la inserción de una dirección (url) para cada evento, siempre y cuando la necesite.

La interacción con el ejercicio de tipo crucigrama o JCross permitió incorporar a la propuesta de solución las opciones de determinar el tamaño de la cuadrícula, la inserción manualmente de las palabras y luego elegir la opción de generar el ejercicio, además, luego de haber creado el crucigrama brinda la posibilidad de adicionar pistas para una mayor comprensión y realización del ejercicio. Para el desarrollo del mismo se hace necesario el empleo de diferentes tecnologías y herramientas, además, del uso de una metodología para guiar el proceso de desarrollo del componente.

1.3 Metodología, tecnologías y herramientas

Actualmente existe una amplia variedad de metodologías, tecnologías y herramientas que permiten la creación y el desarrollo de aplicaciones web. Para el desarrollo de la propuesta de solución no fue necesario realizar un estudio previo de estas, debido a que en la presente investigación se emplea las definidas por la arquitectura del marco de trabajo Xalix. Por lo anteriormente expuesto se decidió hacer uso de la siguiente metodología de desarrollo de software, herramientas y tecnologías. El empleo de las mismas permitió la implementación del componente.

1.3.1 Metodología de desarrollo de software

Las metodologías de desarrollo de *software* son un conjunto de procedimientos, técnicas y ayuda documental para el desarrollo de productos de *software*. La selección de la metodología adecuada, garantiza la creación de un *software* de calidad, en el tiempo planificado y con los costes previamente establecidos. Están divididas en dos grandes grupos: metodologías tradicionales o robustas y metodologías ligeras/ágiles. La diferencia que existen entre ambos grupos es que las metodologías tradicionales son procesos muchos más controlados, con políticas y normas tratando de buscar la calidad del software a través del orden y la documentación, sin embargo, las ágiles son procesos menos controlados y con pocos principios que tratan de buscar la calidad del *software* a partir de la comunicación inmediata y directa entre las personas que intervienen en el proceso de desarrollo. (30)

Atendiendo a esto se propone utilizar una metodología que favorezca el desarrollo rápido de aplicaciones y mantenga una documentación consistente y bien organizada. Debido a que los recursos educativos se desarrollaron empleando la arquitectura del marco de trabajo Xalix que utiliza la metodología de desarrollo Proceso Unificado Ágil (AUP) en su variante para la Universidad, se hace necesario su uso para mantener la uniformidad en la generación de la documentación.

Agile Unified Process

El Proceso Unificado Ágil o Agile Unified Process (AUP) es una versión simplificada del Proceso Unificado de Rational (RUP). Donde se describe una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. (31)

El AUP aplica técnicas ágiles incluyendo:

- ❖ Desarrollo dirigido por pruebas.
- ❖ Modelado ágil.
- ❖ Gestión de cambios ágil.
- ❖ Refactorización de base de datos para mejorar la productividad.

Toda metodología debe ser adaptada según las características de cada proyecto exigiéndose que el proceso sea configurable. De esta manera se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la Universidad de las Ciencias Informáticas (UCI). Esta adaptación establece tres fases de desarrollo las cuales son (31):

- ❖ Inicio
- ❖ Ejecución
- ❖ Cierre

AUP propone 7 disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de vida de los proyectos de la UCI tener 8 disciplinas, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: Modelado de negocio, Requisitos y Análisis y diseño en AUP están unidos en la disciplina Modelo, en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina Implementación, en el caso de Prueba se desagrega en 3 disciplinas: Pruebas Internas, de Liberación y Aceptación y la disciplina Despliegue se considera opcional. Las restantes 3 disciplinas de AUP asociadas a la parte de gestión para la variación UCI se cubren con las áreas de procesos que define CMMI-DEV v1.3 para el nivel 2, serían CM (Gestión de la configuración), PP (Planeación de proyecto) y PMC (Monitoreo y control de proyecto). (31)

1.4 Herramientas y tecnologías

1.4.1 Herramienta CASE

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son un conjunto de programas y ayudas que facilitan la automatización del ciclo de vida del

desarrollo de un Software, apoyando de esta manera a los analistas, ingenieros de software y desarrolladores. Fueron desarrolladas para organizar y estructurar de forma correcta y eficiente un grupo de actividades propias del proceso de desarrollo de software y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de software (32). Para el desarrollo de la propuesta de solución se empleará como herramienta CASE Visual Paradigm en su versión 8.0.

Visual Paradigm v8.0

Es una herramienta que soporta el modelado por UML¹¹ y el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Proporciona características gráficas sencillas que facilitan la realización de los siguientes diagramas: de clases, casos de uso, comunicación, secuencia, estado, entidad-interrelación, actividades y componentes. (33)

Esta herramienta proveer beneficios como:

- ❖ Verificar el uso de todos los elementos en el sistema diseñado.
- ❖ Automatizar el dibujo de diagramas.
- ❖ Ayudar en la documentación del sistema.
- ❖ Ayudar en la creación de relaciones en la Base de Datos.
- ❖ Generar estructuras de código.

1.4.2 Lenguajes para el desarrollo

Un lenguaje de programación es una herramienta que permite crear programas y *software*. Existen dos grandes grupos de lenguajes de programación: los del lado del cliente y los del lado del servidor. En la siguiente investigación se utilizan los dos grupos, los cuales se describen a continuación.

Lenguajes de desarrollo del lado del cliente

Los lenguajes de lado cliente son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pre-tratamiento. En la propuesta de solución se utiliza como lenguaje del lado del cliente HTML, CSS¹² y JavaScript, las cuales permiten el dinamismo de la aplicación en los navegadores, permitiendo de esta forma una rapidez y optimización de los mismos.

¹¹ **UML**: Unified Modeling Language (Lenguaje Unificado de Modelado). Lenguaje de modelado de sistemas de software más conocido en la actualidad.

¹² **CSS**: Cascading Style Sheet. (Hoja de estilo cascada)

HTML 5.0

Lenguaje de Marcado de Hipertexto¹³ (HTML), es el lenguaje básico para la elaboración de páginas web. Define comandos, marcas o etiquetas que permiten delimitar la estructura lógica de un documento web. Ofrece nuevas características con respecto a las versiones anteriores de HTML que proporcionan no solo una amplia compatibilidad de medios, sino también una compatibilidad mejorada para la creación de aplicaciones web que pueden interactuar de una manera más sencilla y efectiva con el usuario, sus datos locales y los servidores. (34)

Entre las principales ventajas que presenta HTML5, tenemos las siguientes:

- ❖ El código es más simple lo que permite hacer páginas más ligeras que se cargan más rápidamente favoreciendo la usabilidad y la indexación en buscadores.
- ❖ Ofrece una compatibilidad mayor con los navegadores de dispositivos móviles.
- ❖ Incluye una nueva etiqueta de dibujo sobre la página web, llamada canvas, que vuelve el proceso de crear dibujos en el sitio web tan fácil como dibujar con aplicaciones como Paint.
- ❖ Ofrece la posibilidad de obtener un código más limpio y fácil de depurar, que los códigos de los estándares anteriores.

CSS3

CSS (Hojas de estilo en cascada) *“...es un lenguaje que trabaja junto con HTML para proveer estilos visuales a los elementos del documento, como tamaño, color, fondo y bordes”* (35). En la propuesta de solución se empleará CSS3 que permite separar los contenidos de su presentación y es imprescindible para la creación de páginas web complejas. En esta versión los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. (36)

Las hojas de estilo pueden compartirse en múltiples páginas, haciendo fácil la actualización de un sitio web completo, simplemente cambiando una línea de la CSS. Proporciona ventajas como almacenamiento en caché, mejor rendimiento, optimiza los tiempos de carga y de tráfico en el servidor. Al haber dividido contenido y apariencia se obtiene archivos más ligeros, y esto reporta dos beneficios: reduce notablemente los tiempos de carga del sitio en el navegador y el volumen de tráfico de nuestro servidor.

¹³ **Hipertexto:** Documento que contiene información cruzada con otros documentos, lo cual permite pasar de un documento al referenciado desde la misma aplicación que se está visualizando.

A continuación se muestran algunas de las ventajas (36):

Ventajas

- ❖ Mayor accesibilidad con una gama más amplia y variada de dispositivos.
- ❖ Mayor facilidad para realizar cambios en el sitio en un solo archivo CSS en lugar de todas las páginas.
- ❖ Archivos más pequeños/descargas más rápidas.
- ❖ Menos código en la página más fácil de interpretar.
- ❖ Permite que los usuarios personalicen el sitio según sus necesidades.
- ❖ Mayor control sobre el código, permite entregar el código en el orden preferido por los lectores de pantalla.

JavaScript v6

JavaScript *“Es un lenguaje interpretado que se emplea principalmente en la construcción de páginas Web, con una sintaxis muy semejante a Java y a C. Pero, al contrario de Java, no se trata de un lenguaje orientado a objetos propiamente dicho, sino que éste está empleando prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad...”*. (37)

Este lenguaje posibilita varias funcionalidades entre las que se encuentran:

- ❖ Interfaces amigables a la vista del usuario.
- ❖ Páginas web más dinámicas.
- ❖ Validaciones de datos del lado del cliente.
- ❖ Actualizaciones de información en tiempo real.

JavaScript v6 brinda un conjunto de librerías que son de gran ayuda en el desarrollo de una aplicación web, entre las más conocidas se encuentra jQuery. Esta librería es de fácil manejo en el tratamiento de los objetos DOM¹⁴ que es una estructura de objetos que genera el navegador cuando se carga un documento y se puede alterar mediante Javascript para cambiar dinámicamente los contenidos y aspecto de la página. (38)

Lenguajes de desarrollo del lado del servidor

Los lenguajes de lado servidor que son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. En la

¹⁴ **DOM:** Document Object Manager. (Administrador de objetos de documento).

propuesta de solución se utiliza como lenguaje del lado del servidor, PHP. Este lenguaje permite desarrollar todo tipo de aplicaciones web y es útil en trabajos donde es necesario acceder a información centralizada.

PHP v5.4.16

PHP es un lenguaje de script interpretado, utilizado para la generación de páginas web dinámicas. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. Al ser un lenguaje libre posee características ideales para la creación de páginas web. (39)

Principales características (39):

- ❖ Brinda soporte para diferentes sistemas gestores de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, SybasemSQL, Informix.
- ❖ Integración con varias bibliotecas externas, permite generar documentos en formato pdf (documentos de Acrobat Reader) hasta analizar código XML¹⁵.
- ❖ Ofrece una solución simple y universal para las paginaciones dinámicas en la web.
- ❖ Fácil de mantener y utilizar.
- ❖ En su versión 5.4.x se han mejorado los mensajes de error de análisis y las advertencias de argumentos incompatibles.
- ❖ <?=¹⁶ ahora está siempre disponible, sin tener en cuenta la opción de *php.ini*.
- ❖ Soportado por una amplia comunidad de desarrolladores, como producto de código abierto, goza de la ayuda de un gran grupo de programadores, permitiendo reparar rápidamente los fallos de funcionamiento que se encuentren.
- ❖ El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar sus propias capacidades.

1.4.3 Framework de desarrollo

Un *framework*¹⁶ de desarrollo es un conjunto de componentes que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web. Los objetivos principales que persigue son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo. (40)

¹⁵ **XML**: eXtensible Markup Language (Language de Marca Extensible). Metalenguaje extensible de etiquetas desarrollado, de marcado sencillo, similar al HTML.

¹⁶ **Framework**: se refiere a una estructura de software compuesta por componentes, los cuales pueden ser personalizados, modificados o cambiados en dependencia de las necesidades de la aplicación a desarrollar.

Symfony v2.7.9

Symfony2 ha sido ideado para aprovechar al máximo las características de PHP. Es un marco de trabajo rápido, flexible y fácil de aprender que le permite a los desarrolladores construir aplicaciones webs más mantenibles. Su arquitectura interna está completamente desacoplada, lo que permite reemplazar o eliminar fácilmente aquellas partes que no se ajustan a un proyecto. (41)

Symfony 2 separa la lógica de negocio, la del servidor y la presentación de la aplicación. Es un marco de trabajo de código abierto que está construido utilizando un contenedor de inyección de dependencias. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Por otra parte, es compatible con la mayoría de gestores de bases de datos. Emplea la arquitectura MVC (Modelo-Vista-Controlador), y proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja (41). Symfony2 contiene una colección de bibliotecas independientes que se pueden utilizar dentro de cualquier proyecto PHP.

Algunas de estas bibliotecas son (41):

- ❖ **HttpFoundation:** contiene las clases Petición y Respuesta, así como otras clases para manejar sesiones y cargar archivos.
- ❖ **Routing:** potente y rápido sistema de enrutado que permite asignar una URI específica a cierta información acerca de cómo se debe manejar dicha petición.
- ❖ **Form:** plataforma flexible para crear formularios y procesar la presentación de los mismos.
- ❖ **Templating:** juego de herramientas para reproducir plantillas, gestiona la herencia de plantillas (es decir, una plantilla está decorada con un diseño) y realiza otras tareas de plantilla comunes.

Symfony2 cuenta con un sistema de bundles, que técnicamente, es un conjunto estructurado de archivos que se encuentran en un directorio y que implementan una sola característica. Cada directorio contiene todo lo relacionado con esa característica, incluyendo archivos PHP, plantillas, hojas de estilo, archivos Javascript, tests y cualquier otra cosa necesaria. (41)

Bootstrap v3.0

Bootstrap es un framework para el desarrollo de aplicaciones de usuario (front-end) rápido y fácil de emplear. Utiliza CSS, HTML y varios recursos (estilos tipográficos, elementos de formulario, botones, tablas, barras de navegación, entre otros). Soporta las versiones más recientes de los navegadores (Firefox, Internet Explorer, Opera, entre otros).

En su versión 3.0 encontramos mejoras en CSS Responsive, eliminación de los layouts fluidos y uso de la actual versión de Glyphicons en lugar de los actuales PNGs. Bootstrap facilita el trabajo con el maquetado

de la página, permite que la plantilla diseñada se pueda observar de la misma manera en todos los navegadores, además brinda la funcionalidad de que el diseño sea optimizado para dispositivos móviles. También en esta versión se realizaron cambios a los Modal, lo que permitió un mejor estilo y comportamiento de los componentes. (42)

JQuery v1.10.2

JQuery es un framework del lenguaje JavaScript, que permite programar sin importar el navegador que utilice cada usuario, ya que funciona de igual manera para la mayoría de los navegadores modernos. Presenta una estructura que le da organización al proyecto y evita la implementación de funcionalidades comunes. Posibilita agregar efectos a las páginas haciéndola más interactiva. Además, que el uso de DOM (*Document Object Model*) y las peticiones AJAX ¹⁷ permiten cambiar el contenido de una página web sin necesidad de actualizarla. (43)

1.4.4 Servidor web

Un servidor web es un programa que gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente. (44)

Apache v2.4.7

Apache v2.4.7 es un servidor web de software libre, código abierto y multiplataforma desarrollado por la *Apache Software Foundation*, puede trabajar en diferentes sistemas operativos y mantener su excelente rendimiento. Además, es gratuito, robusto y destaca por su seguridad. Es utilizado principalmente, para realizar servicio a páginas web, ya sean estáticas o dinámicas. Este servidor se integra con otras aplicaciones, creando el paquete XAMP con Perl, Python, MySQL y PHP. (44)

Principales características (44):

- ❖ Funcionamiento en diferentes plataformas (Windows, Mac OS, Linux).
- ❖ Soporte para el trabajo con lenguajes como PHP, Java, JSP y Perl lo que hace de este servidor web una poderosa herramienta para el desarrollo de páginas web dinámicas.
- ❖ Abundante bibliografía.

¹⁷ **AJAX** es el acrónimo de *Asynchronous Javascript and XML*, JavaScript asíncrono y XML es una técnica que permite la comunicación asíncrona entre un servidor y un navegador en formato XML mediante programas escritos en Javascript.

El servidor Web Apache2 está diseñado para ser un servidor web flexible y potente que puede funcionar sobre varias plataformas y entornos, además de que puede ser personalizado con el objetivo de mejorar las particularidades de cada sitio web.

1.4.5 Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permite a los usuarios la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad (45). En el desarrollo de la propuesta de solución se utilizó como SGBD PostgreSQL en su versión 9.3.

PostgreSQL v 9.3

PostgreSQL es un SGBD relacional, distribuido bajo licencia BSD¹⁸ y con su código fuente disponible libremente. Utiliza un modelo cliente/servidor y usa multiprocesos en lugar de multihilos para garantizar la estabilidad del sistema y funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. A continuación, serán mencionadas las características que permiten comprender, por qué PostgreSQL es el gestor de base datos a utilizar (45):

- ❖ Consistencia en la propiedad, asegurando que las tareas una vez empezadas se finalicen completas y correctamente.
- ❖ Aislamiento en la propiedad, asegura que una operación no puede afectar a las otras.
- ❖ Durabilidad, asegura que una vez realizadas las operaciones, aunque exista fallas en el sistema estas se mantengan persistentes.
- ❖ Funciona en casi todos los sistemas operativos, Linux, Unix, OS, Beos, Windows.
- ❖ Presenta amplia documentación libre, pública y organizada.
- ❖ Adaptable a las necesidades del cliente.
- ❖ Cuenta con varias herramientas gráficas de alta calidad para administrar las bases de datos.
- ❖ Soporta distintos tipos de datos: tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP) y cadenas de bits. También permite la creación de tipos propios.

¹⁸ **BSD**: traducido al inglés como Berkeley Software Distribución es una aplicación que utilice esta licencia permite que otras versiones de la misma pueden tener otros tipos de licencia, tanto propietarias, como BSD o GPL.

PostgreSQL como sistema gestor de base de datos es de fácil uso, además, cuenta con una amplia documentación, estas ventajas resultaron beneficiosas para el desarrollo del componente.

1.4.6 Entorno Integrado de Desarrollo (IDE)

Un Entorno de Desarrollo Integrado (IDE) se define como “*un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica*”. (46)

NetBeans 8.0

El proyecto *NetBeans* es de código abierto y una plataforma de aplicaciones que permiten a los desarrolladores crear aplicaciones web, empresariales, de escritorio y móviles, utilizando la lenguajes como Java, PHP, JavaScript y Ajax y C/C++. Está escrito en Java pero puede servir para otros lenguajes de programación. Es un producto libre y gratuito sin restricciones de uso, disponible para Windows, Mac, Linux y Solaris. (47)

NetBeans en su versión 8.0 presenta un sistema para examinar todos los directorios de cada proyecto, haciendo reconocimiento y carga de clases, métodos y objetos, para acelerar la programación, proponiendo un esqueleto para organizar el código fuente, el editor conjuntamente integra los lenguajes como HTML, JavaScript y CSS, permitiendo la refactorización y búsqueda de usos para CSS y lenguajes de tipo HTML, y el completamiento de código y links para atributos de CSS. (47)

Es robusto, presenta abundante documentación y brinda facilidades para el desarrollo de aplicaciones web utilizando los lenguajes requeridos para el desarrollo de la aplicación. Presenta compatibilidad con el Framework Symfony: proyectos y comandos de Symfony, métodos abreviados de teclado, resaltado de sintaxis PHP en archivos YAML. (47)

1.5 Conclusiones parciales

El estudio realizado sobre el estado del arte relacionado con la creación de recursos educativos a nivel mundial, permitió elaborar el marco teórico que sustenta la investigación. Se describió la metodología de desarrollo de software a utilizar para dar cumplimiento al objetivo general propuesto. Se elaboró los artefactos propuestos por la metodología AUP en su variación para la UCI, facilitando un mayor entendimiento de la propuesta de solución. El empleo de las tecnologías propuestas por la arquitectura del marco de trabajo Xalix como JQuery 1.10.2, HTML5, CSS3, como lenguaje del lado servidor: PHP 5.4.16, como *framework* de desarrollo Symfony en su versión 2.7.9, como servidor web Apache 2.4.7, como SGBD PostgreSQL 9.3, como herramienta CASE Visual Paradigm en su versión 8.0 y el IDE de desarrollo *NetBeans* facilitaron el proceso de diseño e implementación de la propuesta de solución.

Capítulo II. Propuesta de solución

Introducción

El presente capítulo posee como objetivo principal la descripción de las características del componente para la creación de recursos educativos. Se expone de forma minuciosa los requerimientos funcionales y no funcionales. Se define el modelo de dominio, se presentan de igual manera los diagramas de clases del diseño y de secuencia del diseño. La generación de los artefactos, pertenecientes a cada etapa y flujo de trabajo de la metodología de desarrollo seleccionada, permitió sentar las bases para la implementación de la propuesta de solución.

2.1 Descripción del sistema

Como propuesta de solución se desarrolló un componente que tiene como objetivo facilitarle a los usuarios elaborar tres tipos de recursos: línea de tiempo, mapa conceptual y crucigrama. El recurso de tipo línea de tiempo permite que el usuario elaborar una línea de tiempo que cuente con una secuencia de actividades, hechos o eventos de manera ordenada. Facilita obtener una descripción más detallada de la actividad, hecho o evento que se seleccione, permitiendo visualizar su título, descripción, categoría, sitio o lugar donde se desarrolló, posibilita hacer uso de una dirección url en caso que la necesite, una observación sobre otros aspectos y la fecha cuando fue realizada(o) la actividad, hecho o evento.

El mapa conceptual provee a los usuarios la facilidad de relacionar elementos con el fin de ayudar a la comprensión de contenidos. El empleo de cuadro de texto para almacenar información, la conexión mediante líneas, el soporte de plegado y desplegado de los nodos proporciona mayor interactividad con el recurso, brindando la posibilidad de establecer múltiples formas de representación. El recurso posibilita la eliminación de nodos u sus relaciones, además, establecer relaciones bidimensionales entre los nodos.

El recursos de tipo crucigrama permite determinar el tamaño de la cuadrícula, la inserción manualmente de las palabras y luego elegir la opción de generar el ejercicio, además, luego de haber creado el crucigrama brinda la posibilidad de adicionar pistas para una mayor comprensión y realización del ejercicio. La interacción con el recurso de tipo crucigrama garantiza mayor interactividad con el componente, además de incrementar la motivación de los estudiantes en cada contenido, apoyando y evaluando sus habilidades.

2.2 Modelo de dominio

Un modelo de dominio o modelo conceptual muestra clases conceptuales significativas en un dominio. Se utiliza con frecuencia como fuente de inspiración para el diseño de los objetos software (48). A continuación se describen los conceptos que se relacionan con el modelo de dominio.

Descripción del modelo de dominio

Autor: persona que interactúa con la herramienta y que posee el rol de autor.

Recurso educativo (RE): medios empleados por el docente para apoyar, complementar, acompañar o evaluar el proceso de enseñanza-aprendizaje.

Objeto de Información (OI): componente de un recurso educativo.

Metadatos: son datos que describen otros datos, es decir, información relativa a los propios datos que facilitan su catalogación.

Componente: son utilizados para construir una experiencia de aprendizaje desde recursos de aprendizaje reutilizables.

A continuación se muestra la imagen **Figura 1** que representa el modelo de dominio del componente.

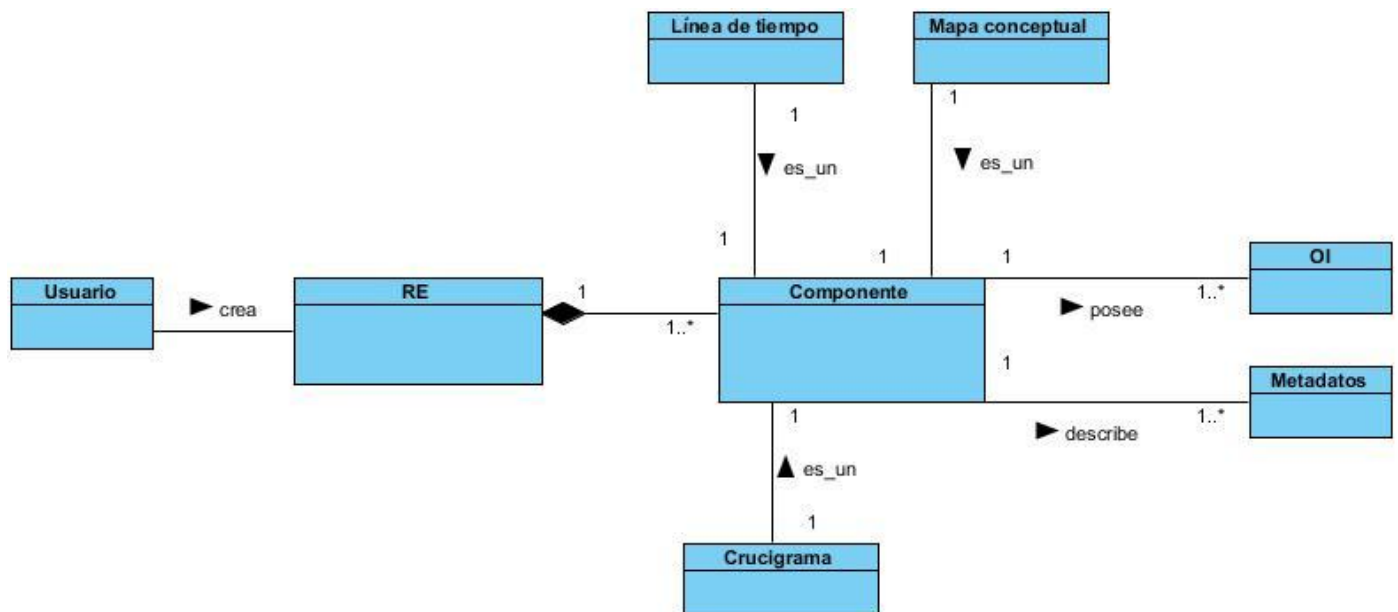


Figura 1. Modelo de dominio.

2.3 Especificación de requisitos

Un requisito de software es la capacidad que debe alcanzar o poseer un sistema o componente de un sistema para satisfacer un contrato, estándar, especificación u otro documento formal (49). El objetivo principal de este acápite es describir los requerimientos funcionales y no funcionales de la propuesta de solución.

2.3.1 Requisitos funcionales

Un requisito funcional especifica una función que un sistema o componente de un sistema debe ser capaz de llevar a cabo. (49)

La propuesta de solución debe satisfacer los siguientes requisitos funcionales:

RF1 Seleccionar línea de tiempo: el componente debe brindar la posibilidad de seleccionar la opción de crear un recurso de tipo línea de tiempo.

RF2 Crear eventos: el componente debe ser capaz de adicionar un evento a la línea de tiempo.

RF3 Ver la línea de tiempo: el componente debe ser capaz de mostrar todos los eventos en forma de línea de tiempo.

RF4 Mostrar los eventos de la línea de tiempo: el componente debe ser capaz de mostrar un listado con todos los eventos.

RF5 Modificar los metadatos básicos de los eventos de la línea de tiempo: el componente debe ser capaz de editar todos los metadatos de los eventos.

RF6 Eliminar los metadatos básicos de los eventos de la línea de tiempo: el componente debe ser capaz de eliminar eventos.

RF7 Seleccionar mapa conceptual: el componente debe brindar la posibilidad de seleccionar la opción de crear un recurso de tipo mapa conceptual.

RF8 Crear conceptos: el componente debe ser capaz de adicionar un concepto al sistema.

RF9 Ver el mapa conceptual: el componente debe ser capaz de mostrar todos los conceptos en forma de mapa conceptual.

RF10 Mostrar los conceptos del mapa conceptual: el componente debe ser capaz de mostrar un listado con todos los conceptos.

RF11 Modificar los metadatos básicos de los conceptos del mapa conceptual: el componente debe ser capaz de editar todos los metadatos de los conceptos.

RF12 Eliminar los metadatos básicos de los conceptos del mapa conceptual: el componente debe ser capaz de eliminar conceptos.

RF13 Seleccionar crucigrama: el componente debe brindar la posibilidad de seleccionar la opción de crear un recurso de tipo crucigrama.

RF14 Crear conceptos: el componente debe ser capaz de adicionar un concepto al sistema.

RF15 Ver crucigrama: el componente debe ser capaz de mostrar un crucigrama y todos los conceptos asociados.

RF16 Mostrar los conceptos del crucigrama: el componente debe ser capaz de mostrar un listado con todos los conceptos del crucigrama.

RF17 Modificar los metadatos básicos de los conceptos del crucigrama: el componente debe ser capaz de editar todos los metadatos de los conceptos del crucigrama.

RF18 Eliminar los metadatos básicos de los conceptos del crucigrama: el componente debe ser capaz de eliminar conceptos del crucigrama.

2.3.2 Requisitos no funcionales

Los requisitos no funcionales en un sistema son aquellos que especifican aspectos técnicos que debe incluir la aplicación. (49)

RNF 1 Usabilidad

- ❖ El componente debe poseer una interfaz fácil de utilizar para cualquier tipo de usuarios con conocimientos básicos de informática y del manejo de ordenadores.

RNF 2 Accesibilidad

- ❖ Las funcionalidades visibles serán las correspondientes según el rol de cada usuario dentro de la aplicación.

RNF 3 Apariencia

- ❖ El componente debe ofrecer una interfaz comprensible para los usuarios con poca experiencia.
- ❖ La interfaz contará con menús que faciliten y aceleren su utilización.
- ❖ Contará con interfaces uniformes y con los mismos colores y diseños.

RNF 4 Disponibilidad

- ❖ La aplicación debe estar disponible en todo momento. Garantizando a los usuarios el acceso a la información.

RNF 5 Soporte

- ❖ La aplicación debe ser implementada bajo tecnología web, la cual permitirá su fácil acceso.
- ❖ Ejecutarse sobre cualquier navegador, siendo como mínimo compatible con:
 - Mozilla Firefox 2.0 y superior.
 - Opera 9.0 y superior.
 - Chrome 4.0 y superior.
- ❖ Restricciones de diseño e implementación.
 - Se aplicará la programación orientada a objetos.
 - El framework de desarrollo que se utilizará es: Symfony v2.7.9
 - Como IDE se empleará NetBeans v8.0.
 - Se empleará como sistema gestor de bases de datos PostgreSQL v9.3.
 - Como servidor web se empleará Apache v2.4.7.

2.4 Historias de usuarios

Las historias de usuarios contemplan de forma sencilla lo que el cliente desea en la aplicación, es decir, este desempeña un papel importante dentro de esta fase. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que se describe. También se utilizan en la fase de pruebas, para verificar si el componente cumple con lo que especifica la historia de usuario. (50)

A continuación se muestran las historias de usuarios del recurso línea de tiempo que guiarán el desarrollo de la solución, el resto de las historias de usuarios de los recursos educativos se describen en el **Anexo1** del presente documento:

Tabla 1. HU_ Seleccionar línea de tiempo.

Historia de usuario	
Número: 1	Nombre: Seleccionar línea de tiempo.
Programador: Luis Antonio Camilo Pérez	Iteración Asignada: 1ra
Prioridad en Negocio: Alta	Tiempo Estimado: 1 día
Riesgo en el desarrollo: 5	Tiempo real: 1 día
<p>Descripción: Permite seleccionar la opción de crear un recurso de tipo línea de tiempo.</p> <p>Objetivo: Permitir crear una línea de tiempo.</p> <p>Acciones para lograr el objetivo (precondiciones y datos): Para seleccionar la opción de crear una línea de tiempo hay que: -Estar autenticado en el componente con el rol X.</p> <p>Comportamientos válidos y no válidos (flujo central y alternos): Debe haberse autenticado el usuario con anterioridad para poder ver la opción de crear un recurso de tipo línea de tiempo.</p> <p>Flujo de la acción a realizar: Cuando el usuario se autentica puede acceder a la opción de crear un recurso de tipo línea de tiempo.</p>	

Observaciones: Para seleccionar la opción de crear una línea de tiempo el usuario debe estar autenticado en el componente.

Tabla 2. HU_Gestionar eventos.

Historia de usuario	
Número: 2	Nombre: Gestionar eventos
Programador: Luis Antonio Camilo Pérez	Iteración Asignada: 1ra
Prioridad en Negocio: Alta	Tiempo Estimado: 5 días
Riesgo en el desarrollo: 1	Tiempo real: 4 días
<p>Descripción: Permite crear, ver, editar, y eliminar los eventos de la línea de tiempo.</p> <p>Objetivo: Permitir gestionar los eventos.</p> <p>Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para seleccionar la opción de crear, ver, editar y eliminar un evento hay que:</p> <ul style="list-style-type: none"> -Estar autenticado en el componente con el rol X. -Debe existir en el componente al menos un evento. <p>Flujo de la acción a realizar: Cuando el usuario selecciona la opción crear, ver, editar o eliminar evento se muestran las diferentes interfaces para gestionar el evento.</p>	
<p>Observaciones: Para gestionar los eventos se deben haber realizado con anterioridad las historias de usuarios número 1,2.</p>	

2.5 Descripción de la arquitectura

La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y

restricciones de la aplicación (51). En la propuesta de solución se hace uso de la arquitectura Modelo Vista Controlador (MVC)

La arquitectura Modelo Vista Controlador en Symfony separa la lógica de negocio (el modelo) y la presentación (la vista), facilitando un mantenimiento sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como en un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo, manteniendo el controlador y el modelo original.

El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El funcionamiento del MVC se basa principalmente en que las vistas obtienen los datos del modelo para mostrarlos al usuario. Cuando un usuario interactúa con la aplicación, ya sea dando clic o escribiendo en la misma, el controlador responde a estas acciones, cambiando los datos en el modelo. Finalmente, el modelo notifica a la vista los cambios ocurridos, de modo que esta pueda actualizar lo que muestra. (52)

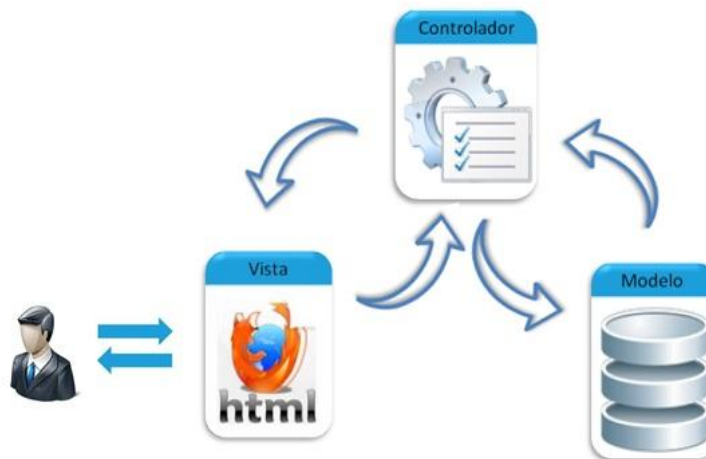


Figura 2. Patrón arquitectónico MVC.

2.6 Patrones de diseño

Los patrones de diseño proporcionan una estructura conocida por todos los programadores. Los patrones no se proponen descubrir ni expresar nuevos principios de la Ingeniería del *Software*, al contrario, intentan codificar el conocimiento, las expresiones y los principios ya existentes. (53) Se definen tres grupos de patrones: de creación, de comportamiento y los estructurales (54), los cuales son conocidos en el mundo

del *software* como patrones GoF¹⁹. Además existen los patrones generales para asignar responsabilidades (GRASP²⁰), que constituyen cuestiones muy básicas, comunes en el diseño. (53)

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. A continuación se explican los patrones GRASP utilizados.

- ❖ **Creador:** realiza el manejo de las peticiones del usuario con la utilización de los métodos *Actions*, en cada una de ellas se pueden crear objetos de las clases que representan las entidades. En el componente se puede apreciar la utilización de este patrón en la clase *Controller*, donde se definen y ejecutan todas las acciones, las clases controladoras son creadoras de dichas entidades.
- ❖ **Experto:** el patrón se encarga de asignar una responsabilidad al experto en información, es decir a la clase que cuenta con la información necesaria para cumplir la responsabilidad. En la propuesta de solución las clases entidades actuarían como expertas ya que poseen toda la información necesaria. (53)
- ❖ **Controlador:** este patrón resuelve el problema de asignar la responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase. Un controlador sirve como intermediario entre una interfaz y la acción que se desee ejecutar (53). En el caso de los recursos desarrollados un ejemplo de clase controladora sería *DefaultController.php*.
- ❖ **Bajo acoplamiento:** un elemento con bajo o débil acoplamiento no depende de demasiados otros elementos (53). Este patrón se evidencia dentro de los recursos en la capa de repositorios ya que las clases de acceso a los datos tienen independencia de las clases de abstracción de datos. El bajo acoplamiento se observa en la separación que existe entre entidad, controlador y las vistas, cada una contiene solo la información necesaria lo cual implica que un cambio en una no afecta a las restantes.
- ❖ **Alta cohesión:** patrón que define que un elemento posea pocas responsabilidades y que estas estén altamente relacionadas (53). Este patrón permite que la implementación de las clases encargadas de ofrecer los servicios de los recursos sean fáciles de comprender, reutilizar, mantener y poco susceptibles a cambios. Se observa en el ejemplo de los controladores donde cada uno contiene únicamente las operaciones e información relacionada a la entidad correspondiente.

Los patrones GOF describen 23 patrones de diseño comúnmente utilizados y de gran aplicabilidad en problemas de diseño usando modelado UML. Se clasifican en tres categorías de acuerdo a su propósito:

¹⁹ **Gof:** Del inglés The Gand of Four (Grupo de Cuatro)

²⁰ **GRASP:** Del inglés General Responsibility Assignment Software Patterns (Patrones Generales para Asignar Responsabilidades)

creacionales, estructurales y de **comportamiento** (35). En el desarrollo del componente se hace uso de un patrón de categoría **estructurales**, los cuales se ocupan de cómo las clases y objetos se combinan para formar grandes estructuras y proporcionar nuevas funcionalidades.

A continuación se dará una explicación de los patrones GOF (36):

- ❖ **Decorador:** patrón que añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad. Este patrón responde a la necesidad de añadir dinámicamente funcionalidad a un objeto (55). Además, de almacenar todo el código HTML usado por todas las páginas de la vista, conocidas como *templates*.

Modelo de diseño

El modelo de diseño crea una representación o modelo de software enfocado en la representación de los datos, las funciones y el comportamiento requerido. Permite al ingeniero de software modelar el sistema o producto que se va a construir permitiendo evaluar su calidad y mejorarlo antes de generar el código, dándole alta calidad al *software*. Con este se obtiene una representación arquitectónica de interfaz y despliegue del sistema. (37)

2.6.1 Diagrama de Clases de Diseño (DCD)

El modelo de diseño es un artefacto ingenieril que incluye diagramas de interacción, de paquetes y de clases. Los diagramas de clases del diseño representan las especificaciones e interfaces de software. (53)
A continuación se presentan los diagramas de clases del diseño Web, el resto de los recursos educativos se describen en el **Anexo2** del presente documento.

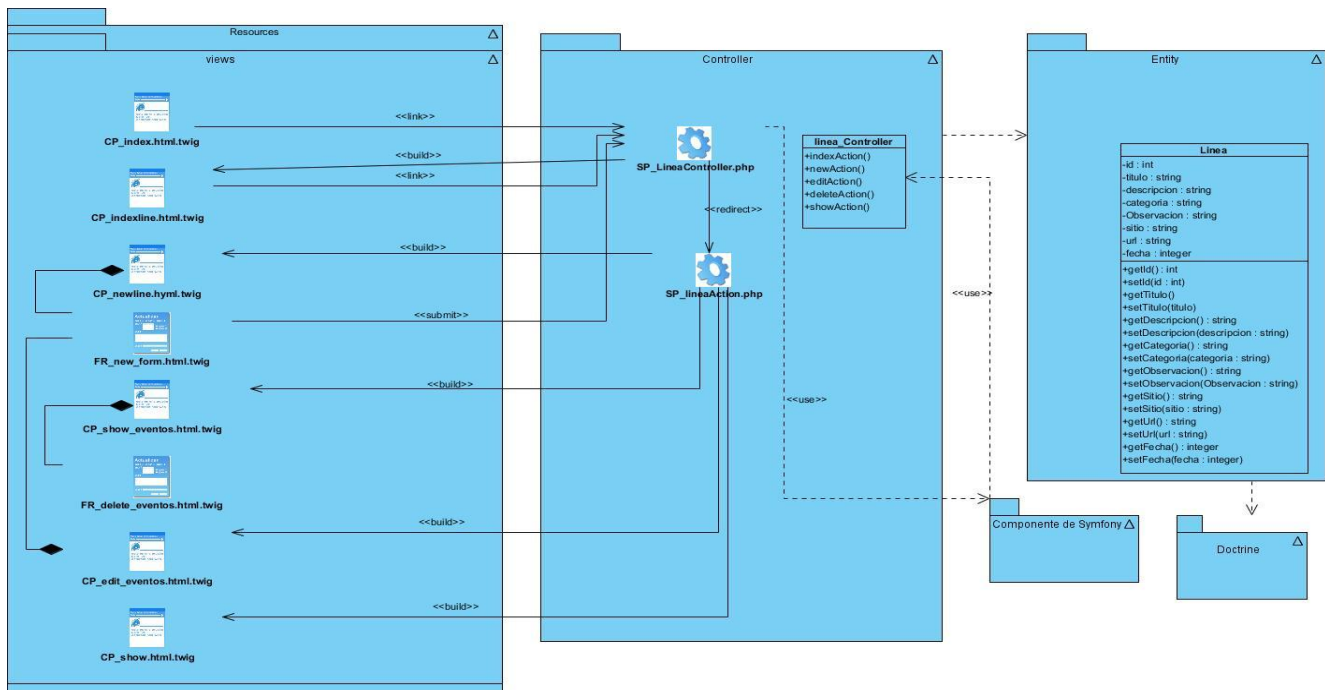


Figura 3. DCD_Línea de tiempo.

2.6.2 Diagramas de secuencia del diseño (DS)

Los diagramas de secuencia que se presentan en la investigación muestran una iteración ordenada según la secuencia temporal de los eventos, es decir muestran los objetos que interactúan y los mensajes que se intercambian ordenados según la secuencia de tiempo en que se realiza cada uno de ellos. (56)

A continuación se muestra el diagrama de secuencia correspondiente al HU Gestionar eventos. Los restantes diagramas se muestran en el **Anexo3**.

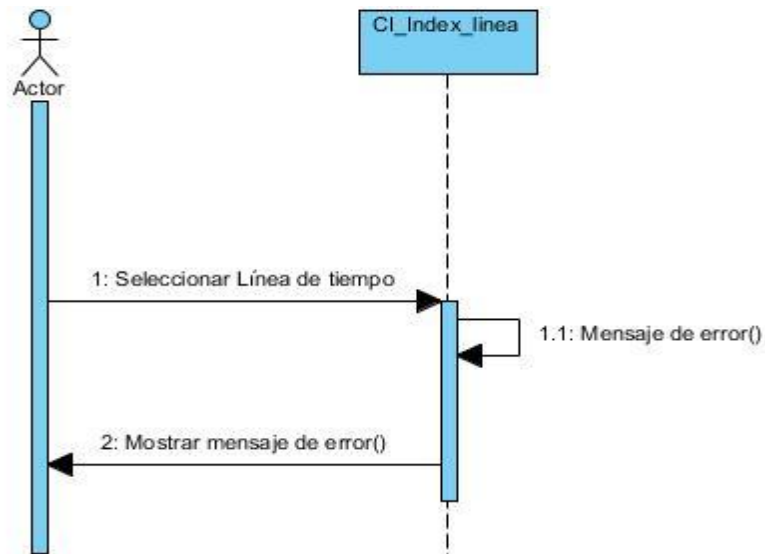


Figura 4. DS_Seleccionar recurso de tipo línea de tiempo.

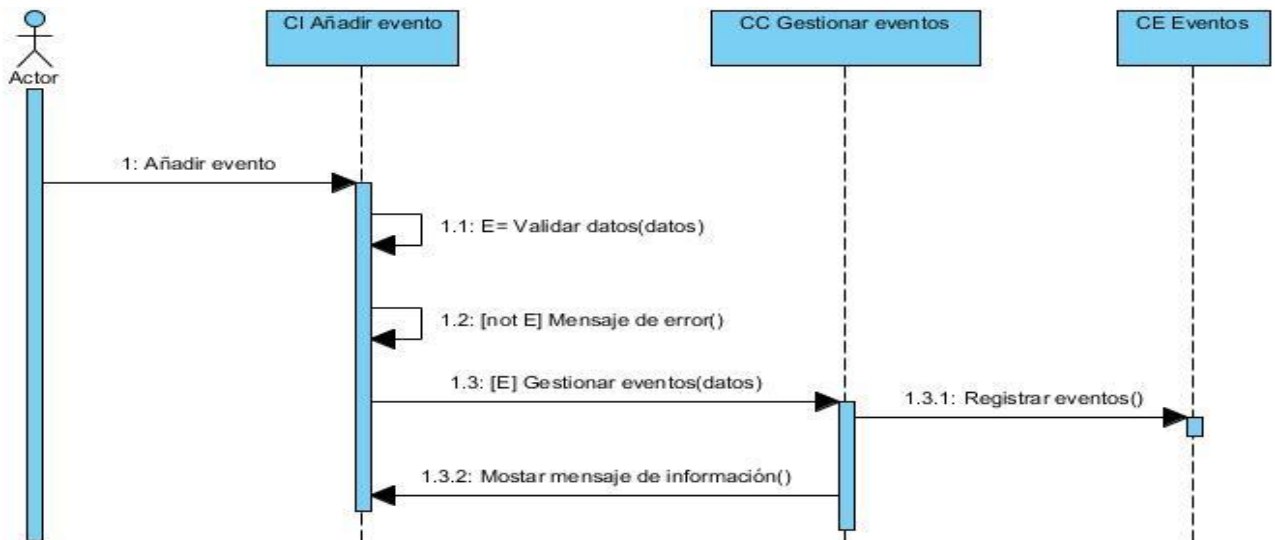


Figura 5. DS_Añadir evento.

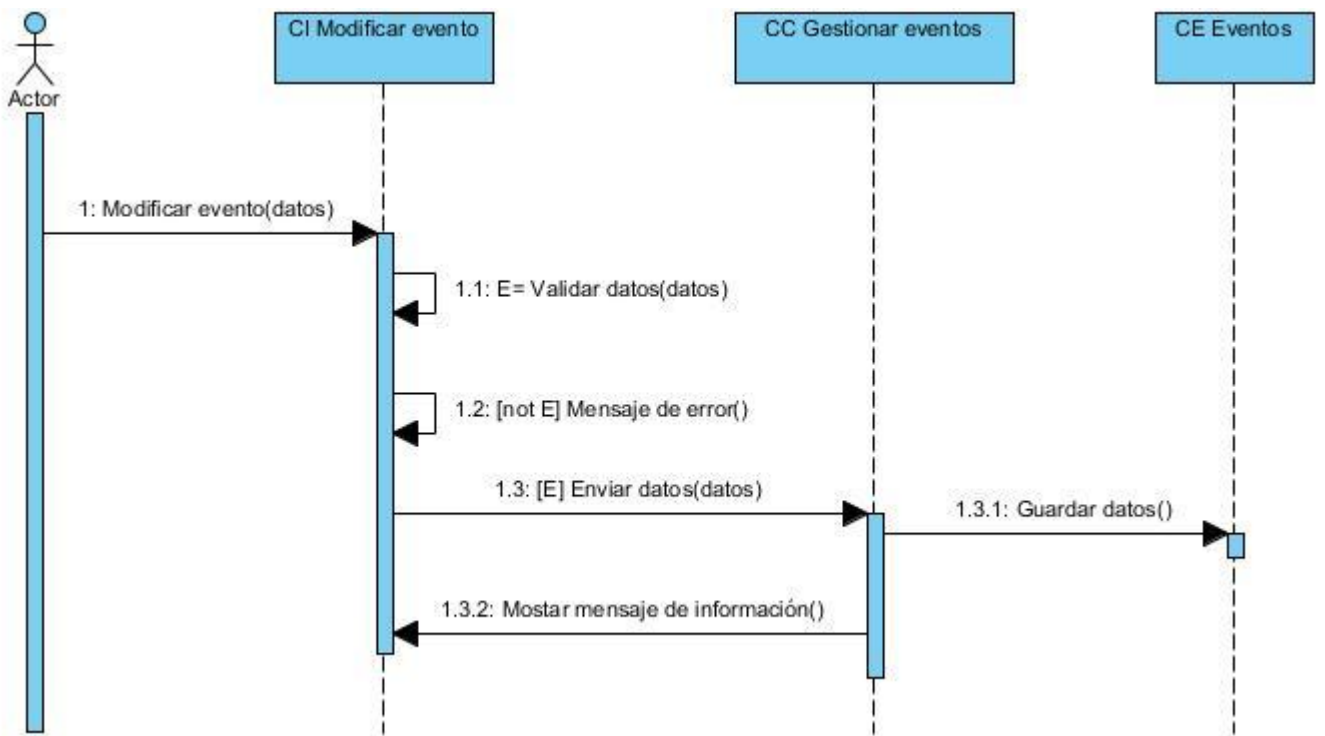


Figura 6. DS_Modificar evento.

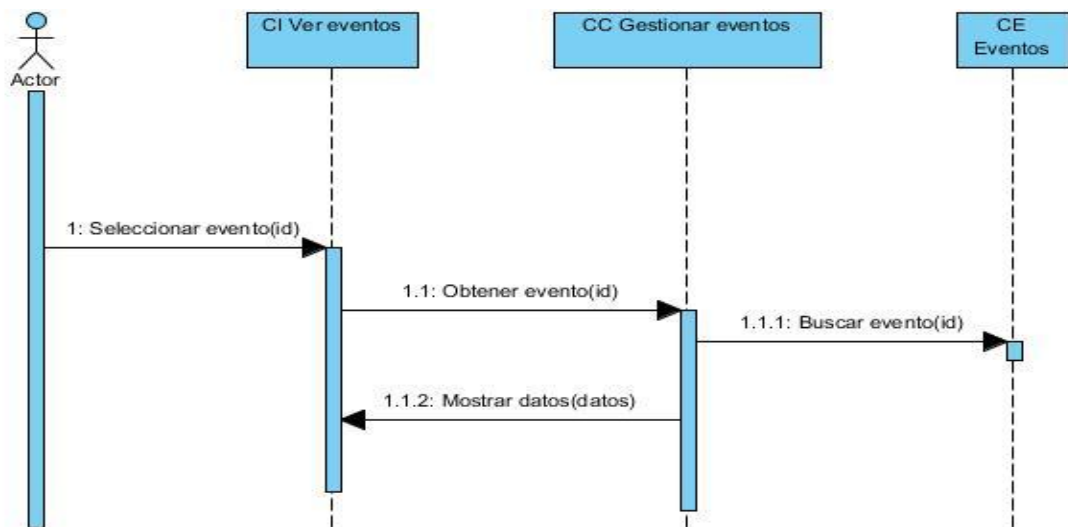


Figura 7. DS_Ver eventos.

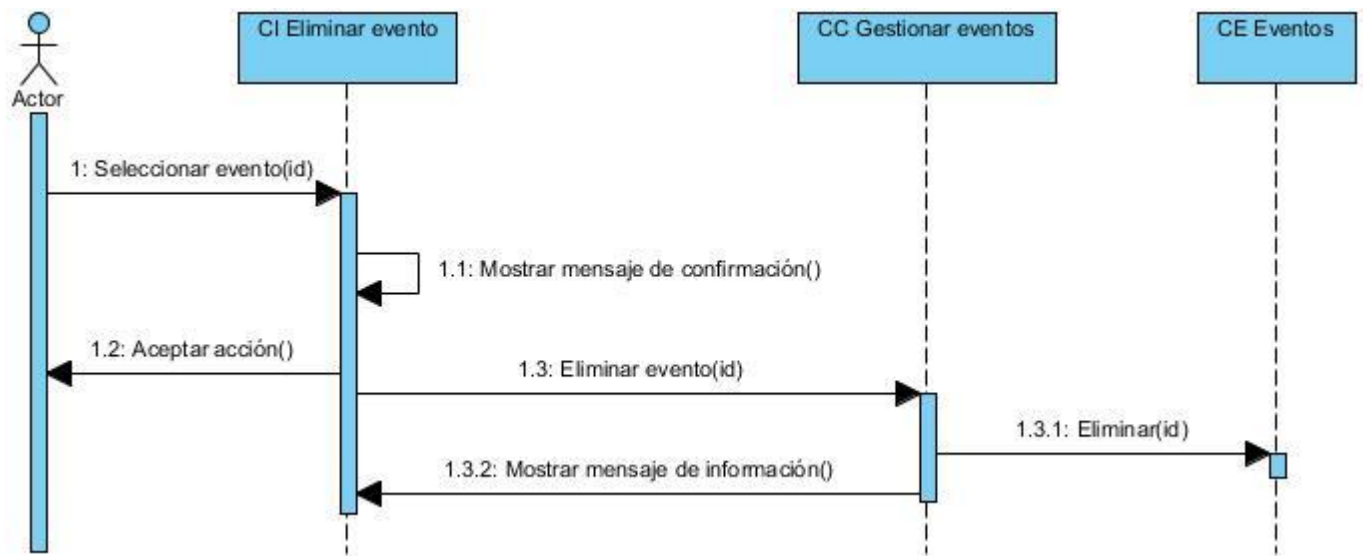


Figura 8. DS_Eliminar eventos.

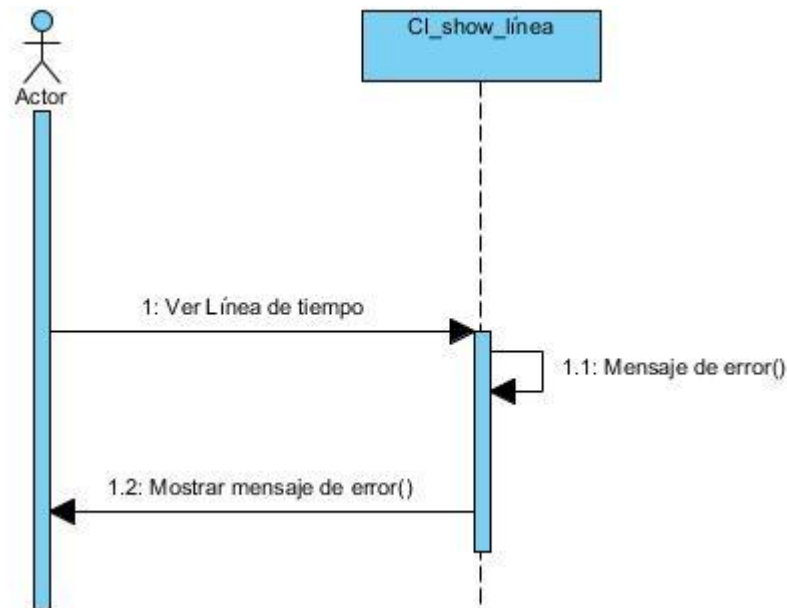


Figura 9. DS_Ver línea de tiempo.

2.7 Modelo de datos

Una base de datos es un almacén que permite guardar información de manera organizada. Una vez obtenidas las clases del diseño se realiza el esquema de la base de datos con el objetivo de asegurar que los datos persistentes sean almacenados eficientemente. (57)

En la **Figura 10** se representa el modelo físico correspondiente a la base de datos de la propuesta de solución.

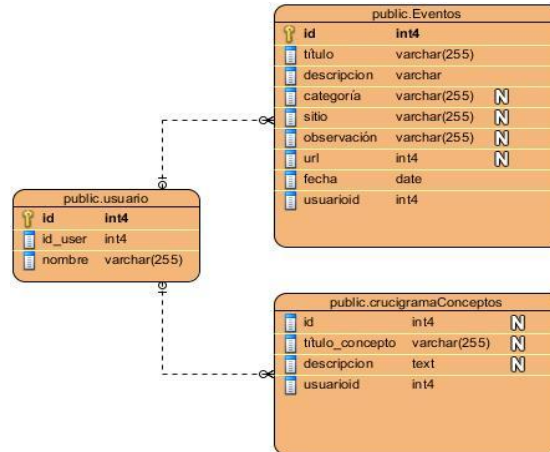


Figura 10. Modelo físico de la base de datos.

Tablas creadas

tb_eventos: es la tabla que almacena los datos sobre los eventos. Cada evento posee un título, una descripción, categoría, sitio/lugar, url y fecha.

tb_crucigramaconceptos: es la tabla que almacena los datos sobre los conceptos asociados al crucigrama. Cada concepto posee un título y una descripción.

2.8 Conclusiones parciales

Los artefactos ingenieriles generados permitieron describir la propuesta de solución, que posteriormente servirá de guía para la implementación. Para ello quedaron definidos los principales conceptos, los cuales se relacionaron en el modelo de dominio, el cual ayudó a comprender el entorno en que se relacionan las funcionalidades. Por su parte los requisitos funcionales y no funcionales permitieron definir las capacidades y las cualidades que deben tener las funcionalidades correspondientes al componente. La arquitectura MVC y los patrones de diseños utilizados (GRASP y GOF) proporcionan un conjunto de buenas prácticas que contribuyen con la obtención de los resultados esperados.

Capítulo III. Implementación y pruebas.

Introducción

Una vez terminado el diseño, se dispone a abordar los temas relacionados con los flujos de trabajo implementación y prueba. Es por esto que en el presente capítulo se desarrolla una descripción de la implementación, generando los diagramas necesario que servirán de apoyo para la creación del componente. Para validar el componente desarrollado se utiliza métodos y estrategias de prueba a partir del diseño y empleo de casos de pruebas, es necesario entrar en un proceso de evaluación, que comprueba su adecuado funcionamiento antes de ser usado por los usuarios finales.

3.1 Modelo de implementación

El modelo de implementación describe como los elementos del modelo de diseño se implementan en términos de componentes, como ficheros de código fuente y ejecutables. Este modelo describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en los lenguaje(s) de programación utilizados y como dependen los componentes uno de los otros. (35)

3.1.1 Diagrama de componentes

El diagrama de componentes muestra cómo está dividido el componente y son empleados para estructurar el modelo de implementación en términos de subsistema de implementación y mostrar las relaciones entre los elementos de implementación. (58)

A continuación se muestra el diagrama de componentes del recurso de línea de tiempo, el resto de los diagramas de los recursos educativos se describen en el **Anexo4** del presente documento.

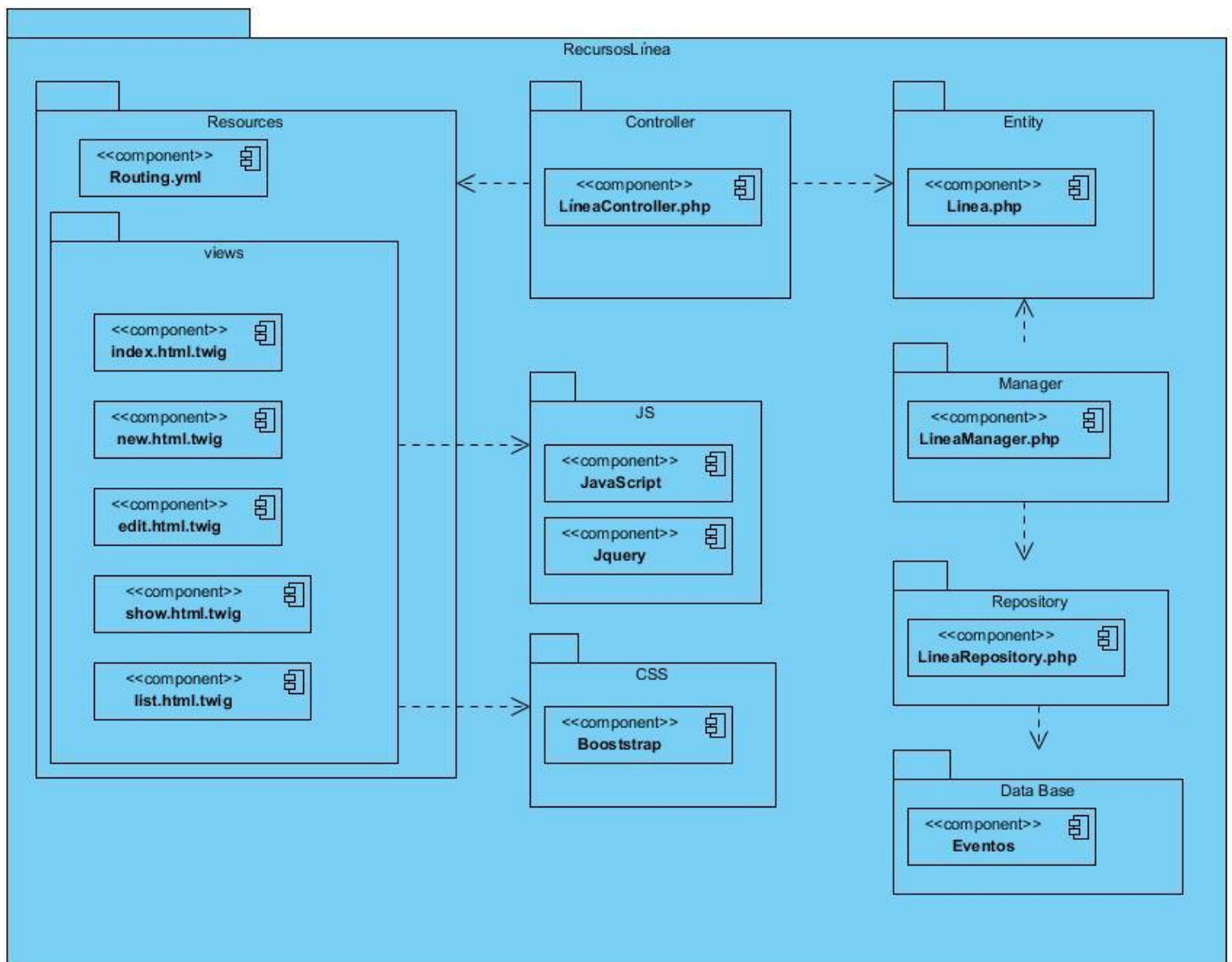


Figura 11. Diagrama de componente de la línea de tiempo.

- ❖ **Paquete views:** contiene los componentes relacionados con las vistas de las funcionalidades del componente.
- ❖ **Paquete Controller:** contiene la clase controladora encargada de gestionar todas las acciones para el paquete de vistas.
- ❖ **Paquete Entity:** contiene las clases entidad generadas por Doctrine que proporciona persistencia a los objetos y un lenguaje de consultas orientado a objetos.
- ❖ **Paquete Manager:** que sirve de fachada a los métodos implementados en el repositorio y a cualquier acción realizada sobre la entidad con el fin de disminuir la cantidad de código necesaria en el controlador.

- ❖ **Paquete Repository:** los repositorios de las entidades deben estar ubicados en un directorio llamado Repository y no directamente dentro de Entity, esta clase posibilita la interacción con la base de datos.
- ❖ **Data Base:** representa la base de dato generada por el componente para almacenar los datos pertenecientes a los eventos de la línea de tiempo.
- ❖ **Resources:** Incluye los archivos de configuración del sistema, además, incluye el paquete de vistas views
- ❖ **JS:** almacena todos los archivos y librerías de JavaScript y JQuery que se utilizan en el componente.
- ❖ **CSS:** almacena todos los archivos css que se utilizan en el componente.

3.1.2 Diagrama de despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación (59). A continuación se muestra el diagrama de despliegue de la propuesta de solución.

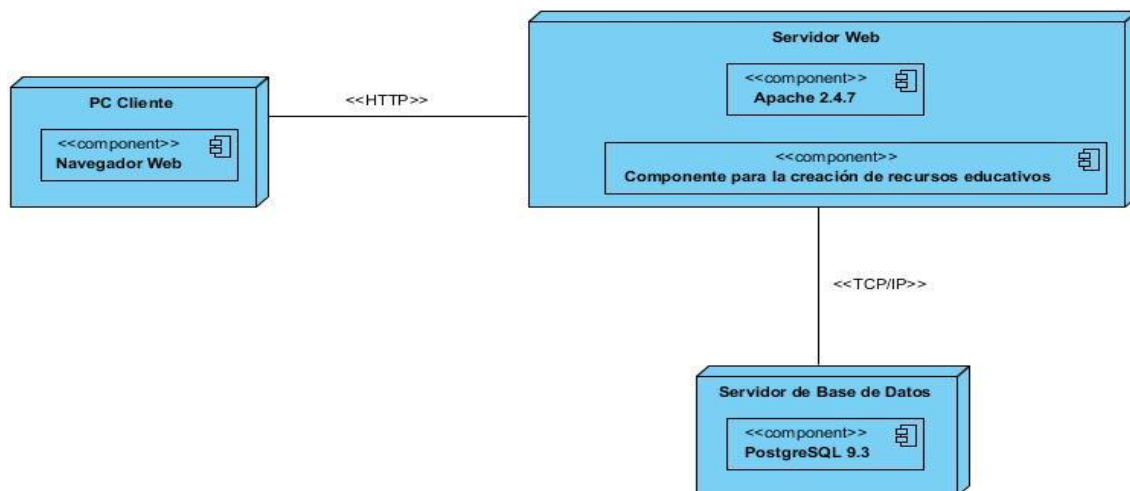


Figura 12. Diagrama de despliegue.

El nodo PC cliente representa los ordenadores que permitirán la interacción del usuario con la aplicación y que establece una comunicación mediante el protocolo HTTP con los nodos donde se encuentra el servidor web. Desde este último se tiene acceso al servidor de base de datos a través del protocolo TCP/IP.

3.2 Estándares de codificación

Con el objetivo de lograr que otros programadores comprendan el código (como identificar las variables, las funciones o métodos) se hace necesario establecer un estándar de codificación, para obtener un producto final con mayor calidad. En general, un estándar de codificación son reglas que se siguen para la escritura del código fuente. (60)

El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad, es establecer un estándar de codificación sobre el que se efectuarán luego revisiones del código de rutinas. (60)

Para implementar el componente, se utilizó el siguiente estándar de código:

- ❖ Un Contenedor de servicios, dado que los servicios son la relación principal, utilizamos la convención de nomenclatura para esta relación.

Ejemplo:

Services:

```

linea.form.linea_type:
  class: Recursos\LineaBundle\Form\lineaType
  arguments: ['@service_container']
  tags:
    - { name: form.type, alias: xalix_linea_type }
linea.form.nomblinea_type:
  class: Recursos\LineaBundle\Form\nomblineaType
  arguments: ['@service_container']
  tags:
    - { name: form.type, alias: xalix_nomblinea_type }

```

- ❖ Un espacio de nombres y la clase completamente calificado debe tener la siguiente estructura `\<Vendor Name>\(<Namespace>)*<Class Name>`

Ejemplo:

```
Recursos\LineaBundle\Entity\Linea
```

- ❖ Cada caracter “_” en el nombre de la clase se convierte en un DIRECTORY_SEPARATOR. El caracter “_” no tiene un significado especial en el espacio de nombre.

Ejemplo:

```

* @Route("/show_line", name="xalixfrontend_showline")
  * @return Response
public function showlineAction()

```

```
{
    $linea=$this->container->get('recursos_linea.manager')-
>getLineaActivity();
    return $this->render(
        'LineaBundle:linea:show.html.twig',
        array(
            'linea' => $linea ) );
}
```

- ❖ Los caracteres alfabéticos en los nombres de los vendors, espacios de nombres y nombres de clases pueden ser cualquier combinación de minúsculas y mayúsculas.

Ejemplo:

```
class LineaController extends Controller
{
```

- ❖ Etiquetas PHP

Todas las etiquetas PHP deben estar completas (`<?php ?>`), no reducidas (`<? ?>`).

- ❖ Los bloques de código deben estar siempre limitados por llaves, excepto en los casos en que tengan una sola línea. Las llaves siempre estarán al mismo nivel de la sentencia de código de la que proceden.

Ejemplo:

```
public function getDescription() {
return $this->description;
}
```

- ❖ Los nombres de las funciones comenzarán siempre con letra minúscula, cuando sean compuestas (tiene más de una palabra).

Ejemplo:

```
nombre_Metodo() {
```

- ❖ Las funciones para obtener algún dato de un objeto específico comenzarán utilizando el verbo inglés *get*, para lograr uniformidad con los métodos del mismo tipo que incorpora el *framework* Symfony2.

Ejemplo:

```
getNombres();
```

- ❖ Las declaraciones de clases abren llaves en la línea siguiente de la declaración, con el mismo nivel de margen de la declaración.

Ejemplo:

```
class LineaController {
```

- ❖ Los comentarios son notas cortas explicativas que se agregan al código para aportar mayor información a las personas que lo leen. Los comentarios de línea deben escribirse de forma clara, en el caso de las funcionalidades significativas o clases se usarán comentarios de bloque al inicio de estos.

Ejemplo:

```
{# Esta funcionalidad devuelve la descripción de un evento #}  
public function getDescription() {
```

- ❖ El objetivo fundamental es nombrar las clases y métodos del módulo de forma estándar. En ambos casos se usará el idioma español para hacer más fácil el entendimiento posterior por cualquier desarrollador, en caso de futuros mantenimientos o inclusión de nuevas funcionalidades, si es un nombre compuesto se separaran las palabras usando *underscore*.

Ejemplo:

```
public function eliminar_EventoAction() {
```

3.3 Pruebas del software

La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final que se centran principalmente en la evaluación o la valoración de la calidad del producto.

El objetivo de las pruebas al componente es comprobar la integración del sistema de información globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de los sistemas de información con los que se comunica. (57)

3.3.1 Niveles de prueba

El proceso de realización de las pruebas está compuesto por una serie de niveles entre los que se encuentran: el nivel de pruebas unitarias, el nivel de pruebas de integración, el nivel de pruebas del sistema y nivel de pruebas de aceptación. Una vez implementado el componente fue sometido a los niveles de pruebas que a continuación se detallan, los cuales ayudaron a la detección de los errores existentes.

Pruebas Unitarias: las pruebas de unidad se concentran en la lógica del procesamiento interno y en las estructuras de datos de los límites de un componente. Se le aplican a cada módulo de un software de manera independiente. El objetivo de las pruebas unitarias es aislar cada parte del programa y mostrar que las partes individuales son correctas. (61)

Las mismas se aplican al componente mediante la ejecución de los casos de pruebas definidos. Se prueba la interfaz del componente para asegurar que la información fluye de forma adecuada hacia y desde la unidad de programa que está siendo probada y se examinan las estructuras de datos locales para asegurar que los datos que se mantienen temporalmente conservan su integridad durante todos los pasos de ejecución del algoritmo. (62)

Durante la realización del proceso de pruebas se decidió aplicar pruebas unitarias empleando el framework de pruebas PHPUnit a los algoritmos que presentan una complejidad considerable dentro de cada uno de los componentes. Estas no se planificaron, ni fueron registrados sus resultados porque se consideró oportuno iniciar la etapa de pruebas unitarias durante el desarrollo.

Pruebas de Integración: a partir del esquema del diseño, los módulos probados se vuelven a probar combinados para probar sus interfaces. Consiste en realizar pruebas para verificar que un gran conjunto de partes del software funcionan juntos. Son las pruebas posteriores a las pruebas unitarias y preceden a las pruebas del sistema. (61)

Pues la principal causa es agruparlos todos de una vez (interacción), debido a que muchas veces se pueden perder los datos de una interfaz, un componente puede tener un efecto adverso sobre los demás, las funciones internas de un componente pueden modificar datos globales que necesiten otros, en fin es muy grande la lista de riesgos que puede surgir al integrar un conjunto de componentes simultáneamente. De aquí la necesidad de probar pequeños segmentos del programa en los que los errores son más fáciles de probar y corregir, además es importante utilizar una correcta estrategia para llevar a cabo la integración de los componentes de un sistema. (62)

Pruebas de aceptación: el desarrollador escribe las pruebas realizadas según la HU seleccionada, para realizar la comprobación y validar las funcionalidades del componente, de esta forma, saber si está apto para ser liberado.

3.3.2 Métodos de prueba

A continuación se describen los métodos aplicados para comprobar el correcto funcionamiento de la propuesta de solución.

Pruebas de Caja Blanca

La prueba de caja blanca, denominada a veces prueba de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que (63):

- ❖ Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- ❖ Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- ❖ Ejecuten todos los bucles en sus límites operacionales.
- ❖ Ejerciten las estructuras internas de datos para asegurar su validez.

Pruebas de Caja Negra

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del *software*. Se aplica específicamente el método de partición equivalente que propone dividir el dominio de entrada del programa en clases de datos a partir de las cuales pueden revisarse los casos de prueba (CP). O sea, permiten al ingeniero del *software* obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. El objetivo es demostrar que las funciones del *software* son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene, saber qué es lo que hace el *software* pero sin entrar en detalles de código, es decir, que es lo que hace, y no cómo lo hace. (57)

Estas pruebas permiten encontrar (57):

- ❖ Funciones incorrectas o ausentes.
- ❖ Errores de interfaz.
- ❖ Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- ❖ Errores de rendimiento.
- ❖ Errores de inicialización y terminación.

3.3.3 Diseño de Casos de prueba

A continuación se muestran los casos de prueba de aceptación realizados a las especificaciones de requisitos expuestas con anterioridad, los restantes CP pueden ser consultados en el **Anexo5**.

Tabla 3. CP_Gestionar eventos

CP Gestionar eventos
Descripción general: El CP inicia cuando el autor accede a la interfaz principal del componente y selecciona la opción crear un recurso de tipo Línea de tiempo, luego decide añadir un evento, editarlo, visualizarlo o eliminarlo. El componente permite salir de la edición en cualquier instante.

Condición de ejecución: Debe existir al menos un evento añadido.											
ID del Escenario	Escenario	Entradas								Respuesta del Sistema	Flujo Central
EC 1.1	El usuario selecciona uno de los tres recursos.	N/A								Muestra un menú en la parte izquierda de la página: <ul style="list-style-type: none"> • Línea de tiempo • Mapa conceptual • Crucigrama 	Recursos
EC 1.2	El usuario selecciona Línea de tiempo.	N/A								Muestra una página donde se representará la línea de tiempo con sus opciones de poder añadir y listar eventos.	Recursos/Línea de tiempo.
EC 1.3	El usuario selecciona la opción que le permite añadir un evento.	N/A								Muestra la página correspondiente para la entrada de los datos del evento.	Recursos/Línea de tiempo/añadir evento.
EC 1.4	El usuario introduce los datos y selecciona la	T	D	C	O	S	U	F	Valida los datos introducidos.	Recursos/Línea de tiempo/Añadi	
		V	V	V	V	V	V	V	Añade un nuevo evento.		

seleccionar la opción Guardar	opción Guardar										r evento.
EC 1.5 Cancelar el envío de los datos del evento.	El usuario selecciona la opción Cancelar.	V	V	V	V	V	V	V	V	Elimina los datos introducidos y permanece en la misma página.	Recursos/Línea de tiempo
EC 1.6 Envío incompleto de los datos.	Existen datos incompletos al llenar los campos.	I	V	V	V	V	V	V	V	Muestra un mensaje de Información y un indicador sobre el campo incompleto.	Recursos/Línea de tiempo/Añadir evento
		V	I	V	V	V	V	V	V		
		V	V	I	V	V	V	V	V		
		V	V	V	I	V	V	V	V		
		V	V	V	V	I	V	V	V		
		V	V	V	V	V	I	V	V		
		V	V	V	V	V	V	I	V		
EC 1.6 Envío incorrecto de los datos.	Existen datos incorrectos al llenar los campos.	I	V	V	V	V	V	V	V	Muestra un mensaje de Información y un indicador sobre el campo incorrecto.	Recursos/Línea de tiempo/Añadir evento.
		V	I	V	V	V	V	V	V		
		V	V	I	V	V	V	V	V		
		V	V	V	I	V	V	V	V		
		V	V	V	V	I	V	V	V		
		V	V	V	V	V	I	V	V		
		V	V	V	V	V	V	I	V		

<p>EC 1.7</p> <p>Selección de la opción Mostrar eventos.</p>	<p>El usuario selecciona la opción listar eventos para acceder a otras opciones.</p>	<p>N/A</p>							<p>Muestra la lista de eventos creados por el usuario. Brinda en cada uno de los eventos las opciones:</p> <ul style="list-style-type: none"> • Editar • Eliminar 	<p>Recursos/Línea de tiempo/Listar eventos.</p>
<p>EC 1.8</p> <p>Selección de la opción que permite editar los datos de los eventos.</p>	<p>El usuario modifica datos del evento.</p>	<p>N/A</p>							<p>Muestra los datos del evento a editar, y brinda la posibilidad de cambiar sus valores introduciendo nuevos.</p> <p>Guardar los datos.</p> <p>Permite Cancelar la operación en cualquier momento.</p>	<p>Recursos/Línea de tiempo/Listar eventos/Editar evento.</p>
<p>EC 1.9</p> <p>Modifica los datos que necesite y selecciona la opción de guardar los datos.</p>	<p>El usuario introduce los datos y selecciona la opción Guardar.</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>Valida los datos introducidos.</p> <p>Modifica los datos del evento.</p>	<p>Recursos/Línea de tiempo/Listar eventos/Editar evento/editar</p>
<p>EC 1.10</p>	<p>El usuario selecciona la</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>V</p>	<p>La operación se cancela.</p>	<p>Recursos/Línea de</p>

Selecciona la opción Cancelar	opción Cancelar										tiempo/Listar eventos.
EC 1.11 Existen datos incompletos.	El usuario selecciona la opción Aceptar sin llenar los campos necesarios.	I	V	V	V	V	V	V	Muestra un indicador sobre los campos vacíos.	Recursos/Línea de tiempo/Listar eventos/Editar evento/editar	
		V	I	V	V	V	V	V			
		V	V	I	V	V	V	V			
		V	V	V	I	V	V	V			
		V	V	V	V	I	V	V			
		V	V	V	V	V	I	V			
		V	V	V	V	V	V	I			
EC 1.12 Existen datos incorrectos.	Existen datos incorrectos al llenar los campos.	I	V	V	V	V	V	V	Muestra un indicador sobre los campos donde se encuentran los datos incorrectos.	Recursos/Línea de tiempo/Listar eventos/Editar evento/editar	
		V	I	V	V	V	V	V			
			V	I	V	V	V	V			
		V	V	V	I	V	V	V			
		V	V	V	V	I	V	V			
		V	V	V	V	V	I	V			
		V	V	V	V	V	V	I			
EC 1.13 Selección de la opción de eliminar un	El usuario selecciona la opción Eliminar.	N/A							Muestra un mensaje de alerta que indica si desea eliminar el evento. Muestra la opción de cancelar la operación.	Recursos/Línea de tiempo/Listar eventos/Eliminar evento.	

evento.				
EC 1.14 Selección de la opción de Eliminar.	El usuario selecciona la opción Eliminar.	N/A	Elimina el evento. Regresa a la vista anterior actualizando los datos.	Recursos/Línea de tiempo/Listar eventos
[Las celdas de la tabla contienen V, I, o N/A. V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.]				

Tabla 4. Variables correspondientes al CP efectuado.

No	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Título (T)	Campo de texto	No	Campo que se refiere al Metadato básico del recurso. Solo se le pueden introducir letras.
2	Descripción (D)	Campo de texto	No	Campo que se refiere al Metadato básico del recurso. Solo se le pueden introducir letras.
3	Categoría (C)	Campo de texto	No	Campo que se refiere al Metadato básico del recurso. Solo se le pueden introducir letras.
4	Observación (O)	Campo de texto	No	Campo que se refiere al Metadato

				básico del recurso. Solo se le pueden introducir letras.
5	Sitio (S)	Campo de texto	No	Campo que se refiere al Metadato básico del recurso. Solo se le pueden introducir letras y números.
6	Url (U)	Campo de texto	No	Campo que se refiere al Metadato básico del recurso. Admite letras, números y otros caracteres.
7	Fecha (F)	Selección	No	Campo que se refiere al Metadato básico del recurso. Solo se le pueden introducir letras y números.

3.3.4 Resultados obtenidos

Con el objetivo de verificar el cumplimiento de los requisitos funcionales establecidos para la presente investigación se hace uso de las **Pruebas de Caja Negra**. Además, se hace uso de los casos de prueba generados durante este flujo de trabajo con el fin de detectar la mayor cantidad de no conformidades posibles en las funcionalidades del componente realizándose tres iteraciones de prueba. Para el seguimiento de todo el proceso de corrección de no conformidades se realiza una tabla, la misma contará con la cantidad de casos de pruebas evaluados.

Tabla 5. Resultados obtenidos

Iteraciones	Cantidad de casos de pruebas	No conformidades detectadas			
		Alta	Media	Baja	Total
Iteración 1	6	4	2	3	9
Iteración 2	6	2	1	2	5
Iteración 3	6	0	0	1	1

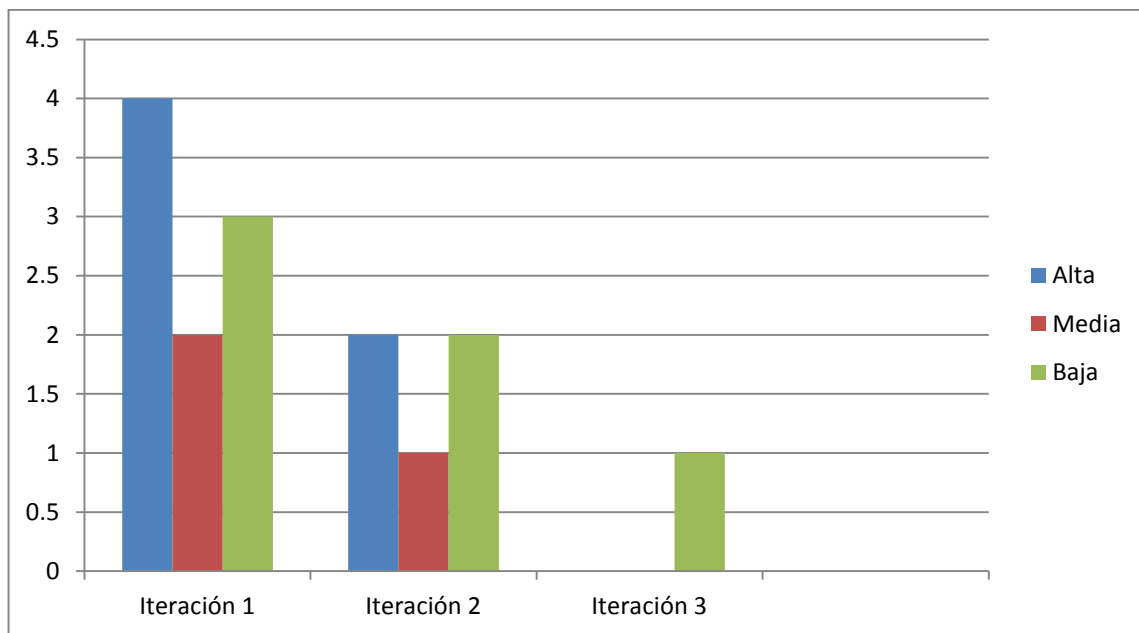


Figura 13. Gráfico de los resultados obtenidos

3.4 Conclusiones parciales

- ❖ El diagrama de componentes generado como parte del modelo de implementación permitió obtener una visión global del funcionamiento de los sistemas, librerías y ficheros del sistema que intervienen en la propuesta de solución.
- ❖ La definición del estándar de codificación, unido a la estructura de carpetas definida por la arquitectura del marco de trabajo Xalix, permitió dotar al código fuente de una mayor organización y claridad.

- ❖ La realización de las pruebas definidas durante el proceso, permitió controlar el proceso de desarrollo del componente con el fin de obtener un producto con la calidad requerida.
- ❖ La aplicación de buenas prácticas demostró su importancia y utilidad durante la implementación de la presente solución informática, a partir de la experiencia descrita en procesos de desarrollo anteriores.

Conclusiones generales

Al término de la investigación se arriba a las siguientes conclusiones:

- ❖ El estudio del estado del arte permitió identificar la poca diversidad de recursos educativos creados en las herramientas desarrolladas en el Centro de Tecnologías para la Formación, evidenciando la necesidad de crear e incorporar nuevos recursos desarrollados mediante tecnologías libres definidas por el marco de trabajo Xalix.
- ❖ El proceso de desarrollo permitió generar los artefactos necesarios y la documentación correspondientes al mismo, sirviendo como base para la implementación del componente.
- ❖ Las pruebas funcionales realizadas permitieron comprobar y validar las funcionalidades del componente desarrollado, quedando satisfechas las necesidades del cliente y cumpliendo con los objetivos planteados.
- ❖ El componente desarrollado permite la creación de recursos educativos de tipo línea de tiempo, mapa conceptual y crucigrama empleando la arquitectura del marco de trabajo Xalix.

Recomendaciones

Con el objetivo de continuar enriqueciendo el funcionamiento del componente desarrollado, se someten a consideración las siguientes recomendaciones:

- ❖ Realizar un estudio de otras herramientas que permiten la creación de recursos educativos en la Web, y de esta manera mejorar el componente desarrollado, a partir de la incorporación de nuevos tipos de recursos educativos.
- ❖ Incorporar nuevos materiales didácticos o funcionalidades con el fin de enriquecer los recursos desarrollados.

Bibliografía

1. **Valverde, J., Garrido, M. C. & Fernández, R.** *Enseñar y aprender con tecnologías: Un modelo teórico para las buenas prácticas con TIC.* 2010.
2. **ZAVANDO, SONIA y FOIX, CRISTIAN.** Estándares e-learning. [En línea] [Citado el: 7 de noviembre de 2015.] <http://docs.google.com/viewer?.booksfactory.com/elearning/docume>.
3. **Grisolia, Maricarmen.** Recursos Didácticos y su importancia en la educación. [En línea] [Citado el: 10 de octubre de 2015.] <http://webdelprofesor.ula.ve/humanidades/marygri/recursos>.
4. **EEES, Espacio interuniversitario de recursos para el.** Espacio interuniversitario de recursos para el EEES. [En línea] [Citado el: 17 de diciembre de 2015.] <http://www.recursoseees.uji.es/>.
5. **Mocoso, P.** *Reflexiones en torno al concepto « Recursos de información».* s.l. : Revista general de información y documentación, 1998. Volumen 8.
6. **A, ARI-ÑO.** *El movimiento open: la creación de un dominio público en la era digital.* España : s.n., 2009.
7. **GONZÁLEZ, ROXANA CAÑIZARES.** Repositorios de recursos educativos para las instituciones de. *Repositorios de recursos educativos para las intituciones de la educación superior.* La Habana : s.n., 2012. 978-84-370-8474-9.
8. **FORTES.** Centro de Tecnologías para la Formación. *Centro de Tecnologías para la Formación.* [En línea] [Citado el: 8 de diciembre de 2015.] <http://gespro.fortes.prod.uci.cu>.
9. **Croda.** Herramienta de autor web CRODA. [En línea] [Citado el: 13 de noviembre de 2015.] <http://croda.uci.cu/>.
10. **ZERA.** Plataforma educativa ZERA. [En línea] [Citado el: 10 de enero de 2016.] <http://zera.uci.cu/>.

11. **Claroline.** Sistema de Gestión de Aprendizaje Claroline. [En línea] [Citado el: 15 de enero de 2016.] <http://www.claroline.net/about-claroline>.
12. **Learning, Exe.** Herramienta de Autor EXeLearning. [En línea] [Citado el: 18 de enero de 2016.] <http://tecnicoenformacion.blogspot.com/2013/03/exelearning7>.
13. **Moodle.** Sistema de Gestión de Aprendizaje Moodle. [En línea] [Citado el: 19 de enero de 2016.] <http://www.moodle.org>.
14. **Hot-potatoes.** Instituto de Tecnologías Educativas. [En línea] [Citado el: 25 de enero de 2016.] www.ite.educacion.es/formacion/materiales/62/cd/modulo_1_primeros_pasos/qu_es_hot_potatoes.html.
15. **González, Rolando Alfredo Hernández León y Sayda Coello.** *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA*. Ciudad de La Habana : s.n., 2011.
16. **Tamajon, Luis Alfonso Garay.** E-Learning y Entornos virtuales de Aprendizaje. 2010.
17. **electrónico, Aprendizaje.** Aprendizaje electrónico. [En línea] [Citado el: 6 de diciembre de 2015.] http://www.ecured.cu/Aprendizaje_electronico.
18. **educativos, Recursos.** Pedagogía y recursos educativos. [En línea] [Citado el: 8 de diciembre de 2015.] <http://www.pedagogia.es/recursos-didacticos/>.
19. **Información, Centro Latino-Americano y del Caribe de.** *POLÍTICA GENERAL DE LA RED DE RECURSOS EDUCACIONALES ABIERTOS*. 2014.
20. **Erika Perrusquia Aguirre, & Tatiana García García.** Evaluación y publicación de contenidos digitales educativos. [En línea] [Citado el: 20 de diciembre de 2015.] <http://bibliotecadigital.conevyt.org.mx/>.
21. **AENOR.** Asociación Española de Normalización y Certificación. [En línea] [Citado el: 20 de diciembre de 2015.] www.aenor.es.
22. **Ochoa, Omar Nájera.** Tipología de Contenido Digital. [En línea] [Citado el: 20 de diciembre de 2015.] <http://portal.educ.ar/debates/educacionytic/inclusion-digital/tipologia-decontenido-digital.php>.

23. **García, Sergio Álvarez.** USO DE CONTENIDOS EDUCATIVOS DIGITALES A TRAVÉS DE SISTEMAS DE GESTIÓN DEL APRENDIZAJE (LMS) Y SU REPERCUSIÓN EN EL ACTO DIDÁCTICO COMUNICATIVO. Madrid : s.n., 2010. 978-84-693-7989-9.
24. **Manjón, Baltasar Fernandez.** Especificaciones y estándares en elearning. [En línea] [Citado el: 22 de diciembre de 2015.] http://reddigital.cnice.mec.es/6/Articulos/articulo_resumen.php?articulo=2.
25. **Marón, José Ramón Hilera González & Rubén Hoya.** Estándares de e-learning: guía de consulta. [En línea] 2010. [Citado el: 22 de diciembre de 2015.] <http://www.cc.uah.es/hilera/GuiaEstandares.pdf>.
26. **Digitales, Criterios para la selección de Recursos.** Ventajas y Desventajas de los Recursos Digitales. [En línea] [Citado el: 10 de enero de 2016.] <https://sites.google.com/site/criteriosrd/criterios-para-eleccion-de-rd/ventajas-y-desventajas-del-uso-de-recursos-digitales>.
27. **Autor, Herramientas de.** Wikipedia. [En línea] [Citado el: 10 de diciembre de 2015.] https://es.wikipedia.org/wiki/Herramienta_de_autor.
28. **Jeudi, Juillet.** Herramienta para crear mapas conceptuales FreeMind. [En línea] [Citado el: 4 de enero de 2016.] <http://recursostic.educacion.es/observatorio/web/fr/home>.
29. **Dipity.** Herramienta para la generación de líneas de tiempo. [En línea] [Citado el: 8 de enero de 2016.] <http://www.eduteka.org/Objetos/Dipity/player.html>.
30. **José H. Canós, Patricio Letelier y Ma Carmen Penadés.** *Métodologías Ágiles en el Desarrollo de Software.*
31. **Sánchez, Tamara Rodríguez.** *Metodología de desarrollo para la Actividad productiva UCI.*
32. **Computación, Departamento de Sistemas Informáticos y.** *Introducción a Herramientas CASE.* Valencia : s.n.

33. **Paradigm, Visual.** Visual modeling tool for building enterprise applications. [En línea] [Citado el: 2 de febrero de 2016.] [http://www.visual-paradigm.com/product/vpuml/features/..](http://www.visual-paradigm.com/product/vpuml/features/)
34. **Programación, Lenguajes de.** Lenguajes de Programación. [En línea] 2010. [Citado el: 05 de febrero de 2015.] <http://www.lenguajes-de-programacion.com>.
35. **GAUCHAT, Juan Diego.** *El gran libro de HTML5, CSS3 y Javascript.* . 2012. 978-84-267-1782-5.
36. **Larman, Craig.** Introducción a CSS . [En línea] 2008. [Citado el: 4 de febrero de 2016.] <http://www.librosweb.es..>
37. **FLANAGAN, David.** JavaScript. La Guía Definitiva. [En línea] 2007. [Citado el: 20 de febrero de 2016.] <http://dialnet.unirioja.es/servlet/libro?codigo=316551>. 978-84-415-2202-2 84-415-2202-2.
38. **JavaScript.** Introducción a JavaScript. [En línea] 2008. [Citado el: 18 de febrero de 2016.] <http://www.librosweb.es..>
39. **php, Características de.** PHP. [En línea] [Citado el: 5 de enero de 2016.] <http://foros.monografias.com/>.
40. **Fabien Potencier, François Zaninotto.** *Symfony, la guía definitiva.* Apress : s.n., 2007.
41. **Eguiluz, Javier.** *Desarrollo web ágil con Symfony 2.* 2011.
42. **Bootstrap.** Libros Web. [En línea] [Citado el: 9 de enero de 2016.] http://www.librosweb.es/que_es_bootstrap.
43. **ALVAREZ, M. A.** Desarrolloweb.com. Manuales. Manual de JQuery. Introducción a JQuery. . [En línea] 2009. [Citado el: 26 de enero de 2016.] <http://www.desarrolloweb.com/articulos/introduccion-jquery.html..>
44. **Apache, Servidor web.** Sitio Oficial de Apache Server. [En línea] [Citado el: 26 de enero de 2016.] http://httpd.apache.org/ABOUT_APACHE.html..

45. **SQL, Postgres.** Página Oficial de Postgres SQL. [En línea] [Citado el: 27 de enero de 2016.] http://www.postgresql.org.es/sobre_postgresql.
46. **desarrollo, Entorno de.** Entorno de desarrollo Integrado. [En línea] 25 de 1 de 2013. [Citado el: 9 de enero de 2016.] <https://fergarcia.wordpress.com/entorno-de-desarrollo-integrado-ide>.
47. **NetBeans.** Entorno Integrado de desarrollo: NetBeans. [En línea] [Citado el: 27 de enero de 2016.] <http://www.netbeans.org>.
48. **Larman, C.** *UML y Patrones. 2ª Edición.* 2003. ISBN: 978-84-205-3438-1..
49. **SÁNCHEZ ALONSO, Salvador, SICILIA URBÁN, Miguel Ángel and RODRÍGUEZ GARCÍA, Daniel.** *Ingeniería del Software: Un enfoque desde la guía SWEBOK.* 2012. 9786077074205.
50. **usuarios, Historias de.** Historias de usuarios. [En línea] [Citado el: 23 de febrero de 2016.] <http://m.genbetadev.com/metodologias-de-programacion/historias-de-usuario-una-forma-natural-de-analisis-funcional>.
51. **Arquitectura de software.** Ecured. [En línea] [Citado el: 8 de febrero de 2016.] http://www.ecured.cu/Arquitectura_de_software.
52. **GREEN, Brad and SESHADRI.** *Shyam O'Reilly Media.* 2013. 9781449355883.
53. **LARMAN, Craig.** *UML Y PATRONES Introducción al análisis y diseño orientado a objetos y al proceso unificado.* Madrid : Prentice Hall, 2003. 8420534382.
54. **GAMMA Erich, HELM Richard, JONHSON Ralph and VLISSIDES John.** *Elements of Reusables Object-Oriented Software.* . 1994. 0-201-63361-2..
55. **Mühlrad, Daniel.** *Patrones de diseño.* 2008.
56. **Xavier Ferré Grau, María Isabel Sánchez Segura (Facultad de Informática–UPM).** *Desarrollo orientado a objetos con UML.* 2004.
57. **Pressman, Roger S.** *Un enfoque práctico.Ingeniería del Software.* España : McGraw -Hil, 2002.

58. **BOOCH Grady, JACOBSON Ivar and RUMBAUGH James.** *EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE.* Madrid : Addison-Wesley, 2000. 84-7829-036-2.
59. **Hualpara, Marca y Quisbert, Hugo Michael.** ANALISIS Y DISEÑO DE SISTEMAS II TRABAJO DE INVESTIGACIÓN Y EXPOSICIÓN. [En línea] 2014. [Citado el: 1 de abril de 2016.]
60. **codificación, Estándares de.** Estándares de codificación. [En línea] [Citado el: 6 de marzo de 2016.] http://librosweb.es/libro/buenas_practicas_symfony/capitulo_4/estandares_de_codigo.
61. **James Rumbaugh, Ivar Jacobson, Grady Booch.** *El Proceso Unificado de Desarrollo Software.* s.l. : Manual de Referencia: Addison Wesley.
62. **S. Pressman, Roger.** *Ingeniería de Software. Un enfoque práctico.* España : Mc Graw Hil, 2005.
63. **Wiley, John.** *Software Engineering Standards.* 2001. 8291983.
64. **e-ABC.** Elearning. [En línea] 2010. [Citado el: 20 de diciembre de 2015.] <http://www.e-abclearning.com/definicion-learning..>
65. **Broche, Orlando Felipe Salvador.** Indicaciones para el trabajo en el marco de trabajo Xalix. La Habana : s.n., 2013.
66. **Valdés, Damián Pérez.** ¿Qué es Javascript? Maestros del Web. . [En línea] 3 de Julio de 2007. [http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/..](http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/)
67. **F. Alonso, L. Martínez, F. Segovia.** *Introducción a la Ingeniería del software.* 2005. ISBN 84-96477-00-2.
68. **CHAUX, HERNANDO RECAMAN.** *ESTUDIO COMPARATIVO DE MARCOS DE TRABAJO PARA EL DESARROLLO DE APLICACIONES WEB DE CÓDIGO ABIERTO.* COLOMBIA : s.n. 319 7900 - 444 7654.
69. **ABIERTOS, POLÍTICA GENERAL DE LA RED DE RECURSOS EDUCACIONALES.** *POLÍTICA GENERAL DE LA RED DE RECURSOS EDUCACIONALES ABIERTOS.* 2014.

70. **Agudelo, M. M.** Plataformas educativas. [En línea] [Citado el: 20 de diciembre de 2015.] <http://aprendeonline.udea.edu.co/banco/html/plataformaseducativas/>.

71. **Torre, Aníbal De la.** Web Educativa 2.0. Edutec. Revista electrónica de tecnología educativa., [En línea] [Citado el: 21 de diciembre de 2015.] <http://www.uib.es/depart/gte/gte/edutece/revelec20/anibal20.htm>.