

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 4



**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

**Módulo para recomendar al usuario cursos y
contenidos en la Plataforma Educativa
ZERA**

Autor: Andy González Rosabal

Tutores: Ing. Miguel Medina Ramírez

Ing. Joel Valdés-Basante Cuza

Ing. Orlando Cruz Torres

“La Habana, junio de 2019

Año 61 de la Revolución”



“...El trabajo llenará gran parte de vuestras vidas y la única manera de sentirse realmente satisfecho es hacer aquello que creéis es un gran trabajo. Y la única forma de hacer un gran trabajo es amar lo que se hace. Si todavía no lo habéis encontrado, seguid buscando. No os detengáis”

Steve Jobs

Declaración de autoría

Declaro ser el autor del presente Trabajo de Diploma y autorizo al Centro de Tecnologías para la Formación (FORTES) de la Facultad 4 en la Universidad de las Ciencias Informáticas, para que hagan el uso que estimen pertinente con el mismo.

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

Andy González Rosabal

Firma del Autor

Ing. Miguel Medina Ramírez

Firma del Tutor

Ing. Orlando Cruz Torres

Firma del Tutor

Ing. Joel Valdés-Basante Cuza

Firma del Tutor

A mis padres, Margarita y Arsenio, con todo mi amor.

Agradecimientos

Al Dios Eterno, por siempre estar conmigo, en todo tiempo y en todo momento, por darme la fe y las fuerzas para terminar los estudios, por ayudarme y protegerme. Gracias Señor.

A mi mamá Margarita, por ser la mejor madre del mundo, por darme la vida, por su inmenso amor, esfuerzo, dedicación, por tenerlo todo listo para que yo no pasara trabajo, por estar conmigo en la risa y en el llanto, por malcriarme como si todavía fuera un niño, por confiar en mí, por alentarme, ayudarme, por todos los días decirme que me adora, por siempre darme todo por mí, ahora yo lo daré todo por ti.

A mi papá Arsenio, por su ayuda, apoyo, esfuerzo y sacrificio, por trabajar duro todos los días para que a mí no me faltara nada, por sus consejos, por enseñarme que el sacrificio de hoy es el beneficio del mañana.

A toda mi familia, a mis abuelas, tías, tíos, primos, gracias por preocuparse por mis estudios, por sus consejos, su ayuda, por siempre creer en mí.

A mis hermanos de San Miguel del Padrón, en especial a la profe Juana por su cariño, por enseñarme a perseverar, a confiar, a creer, por forjar en mí valores como el compromiso y la disciplina.

A mis hermanos de la UCI, nunca olvidaré todos esos lindos momentos, cuando nos veíamos antes de comer a las 7, los martes del fútbol, las celebraciones de cumpleaños, los campamentos, fueron una gran familia para mí; en especial, agradezco a Josue y Sobrino por su ayuda incondicional en la realización de este trabajo.

A mis tutores Miguel, Joel y Orlando, por su ayuda y dedicación.

A todos mis amigos durante estos 5 años, a los compañeros de clase, en especial a Yuliet y a Miguel por acompañarme en los eventos estudiantiles, más que compañeros fueron grandes amigos.

A todos los que de una forma u otra han contribuido con mi formación como profesional y persona.

A todos, muchas gracias.

Resumen

En la Universidad de las Ciencias Informáticas existe la Plataforma Educativa ZERA, desarrollada por el Centro de Tecnologías para la Formación. Dicha plataforma brinda a toda la comunidad universitaria cursos digitales de pregrado, los cuales se imparten en las diferentes facultades como parte del plan de estudios vigente en la universidad. La gran cantidad de cursos y contenidos publicados en ella, provoca que los usuarios al realizar las búsquedas, no siempre obtengan como resultado en las primeras posiciones los ítems más relevantes de acuerdo a sus preferencias y se dificulte la localización de dichos ítems. Con el presente trabajo se obtuvo un módulo recomendador que facilita a los usuarios de la Plataforma Educativa ZERA, el acceso a cursos y contenidos de su interés de acuerdo a su interacción y a la de otros usuarios con la plataforma. El módulo desarrollado responde a las características de los sistemas recomendadores basados en el filtrado colaborativo e implementa el algoritmo de la vecindad próxima para obtener las sugerencias. Para guiar el desarrollo se utilizó la metodología de software AUP-UCI, así como el framework Symfony, el servidor web Apache, entre otras herramientas y tecnologías. Además, se realizaron pruebas funcionales, de carga y estrés y de aceptación, que validaron la calidad y el correcto funcionamiento del módulo implementado.

Palabras clave: *cursos, contenidos, plataforma educativa, módulo recomendador, usuarios.*

Índice

Resumen.....	IV
Introducción	1
Capítulo 1. Fundamentación Teórica.....	5
1.1 Sistemas de Recomendación	5
1.2 Estructura de un Sistema de Recomendación	8
1.3 Clasificación de los sistemas de recomendación	9
1.3.1 Sistemas basados en contenido.....	10
1.3.2 Sistemas basados en filtrado colaborativo	11
1.3.3 Sistemas basados en datos demográficos	12
1.3.4 Sistemas híbridos.....	13
1.4 Clasificación de los algoritmos para el filtrado colaborativo	14
1.5 Aplicaciones de los sistemas de recomendación	19
1.6 Ejemplos de sistemas de recomendaciones	21
1.7 Metodología de desarrollo adoptada.....	23
1.8 Herramientas y tecnologías	25
1.8.1 Entorno de Desarrollo Integrado.....	26
1.8.2 Frameworks de desarrollo	27
1.8.3 Herramienta CASE.....	28
1.8.4 Servidor Web	29
1.9 Lenguajes de programación	30
1.10 Lenguaje de modelado	33
1.11 Sistema Gestor de Base de Datos.....	33
Capítulo 2. Propuesta de solución.....	35
2.1 Modelo conceptual.....	35
2.2 Descripción de la propuesta de solución.....	36
2.3 Especificación de requisitos de software.....	37

2.4 Análisis y diseño	40
2.4.1 Diagrama de clases del análisis	41
2.4.2 Diagrama de colaboración del análisis	41
2.4.3 Arquitectura de software.....	42
2.4.4 Patrón arquitectónico	42
2.4.5 Diagrama de clases del diseño.....	43
2.4.6 Diagrama de secuencia del diseño.....	44
2.4.7 Patrones de diseño	45
2.4.8 Diseño de la base de datos	46
Capítulo 3. Implementación y prueba	49
3.1 Implementación	49
3.1.1 Estándares de código.....	49
3.1.2 Diagrama de componentes	52
3.1.3 Diagrama de despliegue.....	53
3.2 Pruebas de software	55
3.2.1 Pruebas de caja blanca	56
3.2.2 Pruebas de caja negra	56
3.2.3 Diseño de casos de prueba.....	57
3.2.4 Resultados obtenidos	59
Conclusiones generales	63
Recomendaciones	64
Referencias Bibliográficas.....	65

Índice de tablas

Tabla 1. Comparación del KNN con otros algoritmos.	15
Tabla 2. Matriz de votos.	17
Tabla 3. Diferencias entre Metodologías Ágiles y Tradicionales.	23
Tabla 4. Fases de la metodología AUP-UCI.	25
Tabla 5. Historia de usuario del requisito funcional “Valorar un curso”.	40
Tabla 6. Descripción de la entidad “UserInteraction” de la base de datos.	48
Tabla 7. Diseño de casos de prueba para el RF “Configurar notificaciones de tipo recomendación”.	58
Tabla 8. No conformidades detectadas al módulo implementado.	61

Índice de figuras

Figura 1. Sistema de Recomendación. Elaboración propia	6
Figura 2. Estructura de un Sistema de Recomendación. Elaboración propia	8
Figura 3. Clasificación de los Sistemas de Recomendación. Elaboración propia	10
Figura 4. Clasificación de los algoritmos para el filtrado colaborativo. Elaboración propia.	14
Figura 5. Modelo conceptual	35
Figura 6. Diagrama de clases del análisis “Valorar un curso”	41
Figura 7. Diagrama de colaboración del análisis “Valorar un curso”	42
Figura 8. Patrón Modelo Vista Controlador. Elaboración propia	43
Figura 9. Diagrama de clases del diseño.	44
Figura 10. Diagrama de secuencia “Valorar un curso”	44
Figura 11. Modelo Entidad Relación.	47
Figura 12. Definición de la clase AdminOptionsController. Elaboración propia.	50
Figura 13. Definición de la función getNeighbors. Elaboración propia.	50
Figura 14. Definición de variables. Elaboración propia.	50
Figura 15. Definición de comentarios. Elaboración propia.	51
Figura 16. Asignación de variables y llamadas a funciones. Elaboración propia.	52
Figura 17. Estructura de control if else. Elaboración propia.	52
Figura 18. Diagrama de componentes.	53
Figura 19. Diagrama de despliegue. Elaboración propia.	54

Índice de gráficos

Gráfico 1. Cantidad de no conformidades identificadas por las pruebas funcionales.....	59
Gráfico 2. Resultados obtenidos para la prueba de carga.....	60
Gráfico 3. Resultados obtenidos para la prueba de estrés.....	61

Introducción

La constante evolución y el desenfrenado desarrollo de la web, ha transformado la red de redes en un amplio espacio para la búsqueda de información en cualquier formato digital (texto, audio, video, animación, imagen). A menudo, esta se encuentra poco estructurada u organizada, por lo que encontrar información relevante, clasificarla y corregirla, requiere de tiempo, espacio y se hace más difícil para los usuarios localizar la información de su interés individual producto a la sobrecarga existente en la red. (1)

Con frecuencia, es necesario tomar decisiones sin tener la experiencia suficiente sobre las alternativas a elegir. Por ello, día a día las personas confían y tienen en cuenta las recomendaciones de terceros para llevar a cabo sus decisiones personales con mayor provecho e interés. En el entorno informático, el hecho de sugerir es realizado por los sistemas de recomendación. Estos tratan de ser una alternativa al proceso social de recomendación. Generalmente son decisiones sencillas como qué libro leer, qué música escuchar, qué contenido multimedia ver o qué lugares visitar por citar algunos ejemplos. (2)

En general, los sistemas de recomendación se han enfocado en sugerir a los usuarios contenido específico que pueden encontrar en la red. Esto es posible mediante valoraciones que haga el propio usuario sobre lo que prefiere o recomendaciones del sistema sobre lo que prefieren usuarios con características similares. La falta de experiencia o conocimiento sobre las alternativas, así como el gran incremento de la información que genera la misma sociedad, hace que cada día cobre mayor importancia el hecho de tener métodos automáticos o semiautomáticos de filtrado y/o selección de información para la toma de decisiones. (1)

Por otra parte, uno de los métodos de filtrado de información creado para ayudar a los usuarios a resolver el problema de la sobrecarga de información, son los sistemas de recomendación o sistemas recomendadores; los cuales son muy utilizados en campos como el comercio electrónico (e-commerce), el aprendizaje automático (e-learning) y el ocio. (3)

En la actualidad, una de las herramientas de e-learning más usadas son las plataformas educativas, las cuales nacen en el contexto de las nuevas necesidades educativas para responder de forma integral a los requerimientos que impone la sociedad del conocimiento. Se definen como un sitio web, que permite a un profesor tener un espacio virtual donde puede colocar todos los materiales de su curso y otros recursos que considere necesarios con el propósito que sus estudiantes logren alcanzar los objetivos propuestos (4). Poseen gran cantidad de cursos educativos, lo que conlleva a los usuarios realizar una ardua tarea a la hora de encontrar el curso de su preferencia. Muchas son las ocasiones que se desconoce la existencia de cierto curso en particular, o de aquellos cursos que en un momento dado pudieran ser de

interés para algún usuario. Por consiguiente, se pierde la oportunidad de enriquecer los conocimientos de los usuarios en temas que pudieran ser de su interés.

En la Universidad de las Ciencias Informáticas (UCI) se cuenta actualmente con varios centros productivos, destacándose el Centro de Tecnologías para la Formación (FORTES), el cual desarrolla diversos proyectos con el objetivo de ayudar a los docentes a adaptarse a las nuevas normas y personalizar el aprendizaje a partir de nuevas experiencias de aprendizaje digital. Entre los proyectos llevados a cabo por dicho centro, se encuentra la Plataforma Educativa ZERA, la cual se creó como un espacio de apoyo al sistema de enseñanza-aprendizaje en la UCI. Pone a disposición de toda la comunidad universitaria cursos digitales de pregrado y postgrado, propicia un lugar para el trabajo metodológico colaborativo con el objetivo de orientar y elevar la calidad de los cursos, brinda la posibilidad de crear, visualizar e interactuar con contenidos y recursos educativos.

La Plataforma Educativa ZERA, al no contar con un método o servicio que filtre los cursos y contenidos más relevantes para el usuario, hace que sea muy difícil encontrar con facilidad dicha información. Los cursos más relevantes al usuario no siempre están ubicados en las primeras posiciones de los resultados de búsqueda y los contenidos no se pueden obtener con el buscador de la plataforma, por lo que se hace difícil la localización de los mismos; ocupa tiempo y espacio, además de provocar estrés y cansancio a la hora de encontrar los elementos deseados. Todo esto conlleva que, en ocasiones, los usuarios abandonen la búsqueda sin tener el resultado querido.

Por otra parte, el sistema actual no registra en bases de datos la interacción de los usuarios con la plataforma, lo que provoca que se desconozcan los elementos preferidos por los usuarios. Además, el sistema no cuenta con un mecanismo que le brinde al usuario de forma automática un listado con los cursos y contenidos que pudieran resultarle de mayor interés según su interacción y la de otros usuarios con la plataforma.

En correspondencia con lo descrito anteriormente se plantea el siguiente **problema a resolver**: ¿Cómo facilitar a los usuarios de la Plataforma Educativa ZERA, el acceso a los cursos y contenidos de interés de acuerdo a su interacción con el sistema?

El problema anterior determina que el **objeto de estudio** sea: el proceso de recomendación en sistemas informáticos, el cual se enmarca en el **campo de acción**: sistemas de recomendación en plataformas educativas.

Para resolver el problema identificado, se define como **objetivo general**: desarrollar un módulo de recomendación que facilite a usuarios de la Plataforma Educativa ZERA, el acceso a cursos y contenidos de su interés de acuerdo a su interacción y a la de otros usuarios con el sistema.

Para darle solución al objetivo general se trazaron los siguientes **objetivos específicos**:

1. Establecer el marco teórico-conceptual para fundamentar la investigación a partir del estudio de los sistemas de recomendación y las plataformas educativas.
2. Realizar el análisis y diseño del módulo a partir de la metodología de desarrollo de software utilizada.
3. Implementar un módulo que permita recomendar al usuario cursos y contenidos en la Plataforma Educativa ZERA.
4. Ejecutar pruebas de calidad al módulo implementado.

Se define como **idea a defender**: el desarrollo de un módulo de recomendación de cursos y contenidos en la Plataforma Educativa ZERA, permitirá orientar al usuario respecto a los cursos y contenidos que pudieran ser de su interés de acuerdo a su interacción con el sistema.

Con el fin de resolver y dar cumplimiento a los objetivos específicos se emplearon los siguientes **métodos de investigación**:

Métodos Teóricos

Método Histórico-Lógico

Posibilitó hacer un estudio de los fundamentos teóricos de los sistemas de recomendación desde sus inicios hasta la actualidad en el ámbito nacional e internacional, así como las tendencias actuales de estos sistemas.

Método Analítico-Sintético

Se aplicó para todas las fuentes de información: documentos, artículos, trabajos científicos e investigaciones y literatura científica en general. Posteriormente se realizó un análisis de los sistemas de recomendación, sus características, funcionamiento y clasificación. De esta forma se logró entender el tema, se hizo fácil su estudio y se trazaron estrategias para llegar al resultado final.

Método de Modelación

La utilización de este método posibilitó la obtención de los principales artefactos de cada una de las fases de desarrollo del sistema.

Métodos Empíricos

Observación

Se utilizó para apreciar el comportamiento de diferentes plataformas virtuales que hacen uso de sistemas de recomendación como son: Coursera, Miríada X, Floqq, Tutellus y EDUTEKA. Además, la observación permitió enfocar mejor el diseño del sistema y evitar errores.

Consultas de fuentes de Información

Se utilizó para la consulta de las fuentes bibliográficas durante la investigación, lo cual fue esencial para la elaboración del marco teórico.

El presente trabajo de diploma se encuentra estructurado en tres capítulos:

Capítulo 1. Fundamentación Teórica: se hace referencia a los aspectos teóricos que sustentan la base de la investigación realizada. Se describen los conceptos principales para comprender el problema, así como las tecnologías y herramientas a utilizar para desarrollar un módulo que permita recomendar al usuario cursos y contenidos en la Plataforma Educativa ZERA.

Capítulo 2. Propuesta de solución: se exponen las características del sistema, incluidos los requisitos funcionales y no funcionales, se realiza la descripción de la solución, además de los patrones de diseño que se utilizaron y artefactos que propone la metodología escogida para el desarrollo.

Capítulo 3. Implementación y prueba: se validan los requisitos y el diseño del sistema, se especifica el estándar de codificación empleado en la implementación, se obtienen los casos de prueba que se aplican a este, con el fin de validar la solución dada, además, se realiza un análisis de los resultados obtenidos para determinar la implicación de los mismos en el posterior uso del sistema.

Capítulo 1. Fundamentación Teórica

En el ámbito educacional existe gran cantidad y diversidad de material que puede contribuir al proceso de enseñanza-aprendizaje. Con el desarrollo de la web y su utilización a nivel mundial, se tiene una gran cantidad de material útil e interesante para ser empleado tanto por un alumno que desea aprender un tema, como por un docente que quiera preparar algún material didáctico. A través de buscadores, la web se ha convertido en una herramienta fundamental para la recuperación de este tipo de material, pero no siempre el resultado obtenido es el esperado por el usuario. En los últimos años, los sistemas recomendadores han surgido para ayudar al usuario a resolver este tipo de problema, puesto que son capaces de seleccionar, de forma automática y personalizada, el material que mejor se adapte a las preferencias o necesidades de este.

El objetivo de este capítulo es identificar los conceptos y elementos teóricos que permitan comprender las características principales de un sistema de recomendación, factores necesarios además para la realización y descripción del mismo. También se realiza un estudio de la clasificación y estructura de estos sistemas, así como las aplicaciones que tienen en esferas como el comercio electrónico y el aprendizaje automático. Por último, se describen las principales metodologías y herramientas utilizadas en la implementación de la solución propuesta.

1.1 Sistemas de Recomendación

Origen

El origen de este tipo de sistema se remonta a principios de la década de los 90, servicios de filtrado de noticias que permitían a su comunidad de usuarios acceder exclusivamente a aquellas noticias que potencialmente podían ser de su interés. El primer sistema que surgió en este ámbito fue el llamado Tapestry, sistema que permite almacenar la retroalimentación de los usuarios sobre artículos o noticias interesantes para ellos, donde estos artículos o noticias pueden ser utilizados por otros usuarios para determinar si la información del documento es relevante o no. (5)

Otra de las primeras formas de filtrado de información electrónica apareció con el trabajo de Edward M. Housman y Elaine D. Kaskela, en el que se diseñó un método que de forma automática se pudiera mantener a los científicos informados sobre nuevos documentos publicados en sus áreas de trabajo o especialización. Este método se basaba en la creación de un perfil de usuario que contenía ciertas palabras clave relevantes para el usuario, que son utilizadas para buscar coincidencias entre estas y los nuevos documentos o artículos, con el fin de intentar predecir cuáles de las informaciones encontradas serían del interés para los científicos. (5)

Ya en 1997, Paul Resnick y Hal R. Varian (pioneros en la implementación y estudio de los primeros sistemas de recomendación) proponen llamar a estas técnicas con el nombre de Sistema de recomendación por dos razones: en primer lugar, porque puede ocurrir que los usuarios no colaboren explícitamente entre ellos y, en segundo lugar, porque el sistema puede sugerir elementos no conocidos hasta el momento por el usuario y en ese caso se lleva a cabo una recomendación. (6)

La amplia utilización y aceptación de este nuevo paradigma, ha permitido que se pueda confirmar el cierre de la época de la información para dar apertura a la etapa de la recomendación. (7)

Definición

Un primer concepto enunciado por Paul Resnick y Hal Varian en los tiempos del surgimiento de los Sistemas de Recomendación (SR) enuncia que: es aquel sistema que utiliza las opiniones de los usuarios de una comunidad para ayudar a usuarios de esa comunidad a encontrar contenidos de su gusto entre un conjunto sobrecargado de posibles elecciones. (8)

Otro concepto dado por Ricardo Moya refleja que: un SR es un sistema inteligente que proporciona a los usuarios una serie de sugerencias personalizadas (recomendaciones) sobre un determinado tipo de elementos (ítems). Los sistemas de recomendación estudian las características de cada usuario y mediante un procesamiento de los datos, encuentra un subconjunto de ítems que puedan resultar de interés para el usuario. (9) Véase **Figura 1**.

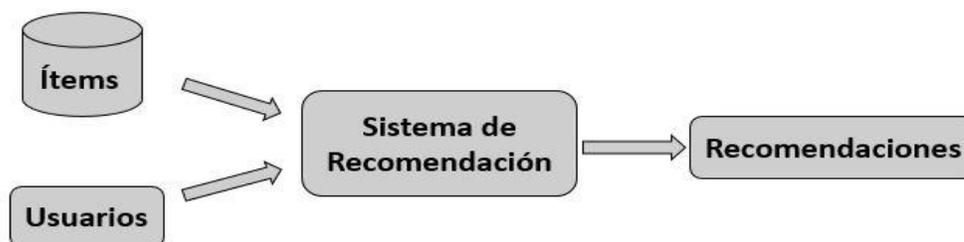


Figura 1. Sistema de Recomendación. Elaboración propia

“Uno de los principales retos que tienen que afrontar los Sistemas de Recomendación es la gestión eficaz del enorme volumen de datos que almacenan, con el objetivo de facilitar a los consumidores la información que satisfaga sus necesidades de una manera rápida y sencilla. Esta necesidad obtiene una mayor importancia en una sociedad como la actual, donde el nivel de exigencia de los usuarios es cada vez mayor”. (10)

Los sistemas de recomendación son herramientas asistidas por ordenadores para la toma de decisiones que proporcionan a los usuarios suficiente información y los ayudan a encontrar lo que buscan o sugieren temas en los que pueden estar interesados. (11)

Los sistemas de recomendación se encargan de proporcionar a los usuarios consejos e información personalizada sobre productos o servicios que puedan ser de interés a la hora de tomar una decisión. Este proceso, en el que el sistema guía al usuario a la hora de realizar una elección, puede proporcionar resultados que sean de gran utilidad, contribuye al ahorro tiempo, proporciona datos relevantes de forma cómoda y fácil, e incluso, se obtiene información que permite valorar opciones que de otra forma antes no se habrían contemplado, algo muy apreciado por la mayoría de los usuarios. (12)

Una vez estudiados los conceptos anteriores, se ha definido en la presente investigación que un SR es aquel sistema que tiene en cuenta los gustos de una comunidad de usuarios sobre determinados ítems, con el objetivo de ayudar a otros usuarios de esa misma comunidad a elegir los ítems de su preferencia.

El problema de la recomendación

El principio de funcionamiento de los sistemas de recomendación está resumido en el problema de la recomendación (13), el cual puede ser formulado de la siguiente forma:

Sea C un conjunto de usuarios, y S el conjunto de todos los ítems posibles a ser recomendados, tales como libros, películas o restaurantes, donde la cardinalidad de ambos conjuntos puede ser grande. Sea además u una función que mide la utilidad del ítem s al usuario c , o sea $u: C \times S \rightarrow R$, donde R es un orden total (enteros no negativos o números reales en un determinado rango). Con estos elementos, se desea, para cada usuario $c \in C$, aquel ítem $s' \in S$ que maximice la utilidad al usuario. Más formalmente:

$$\forall c \in C, s'_c = \operatorname{argmax}_{s \in S} u(c, s)$$

El modelo del proceso de recomendación

Dentro de la concepción de sistemas de recomendación, ocupan un papel central el buscador de recomendaciones, el proveedor de preferencias y el sistema recomendador propiamente dicho. El buscador de recomendaciones (generalmente un individuo) puede solicitar sugerencias al sistema recomendador, o este puede automáticamente ofrecerlas sin una solicitud previa. Asimismo, para la construcción de las recomendaciones, el sistema puede capturar de manera directa las preferencias específicas de los individuos, o puede inferirlas a través de preguntas indirectas. Una vez recopilados todos los datos sobre los usuarios, el sistema está listo para recomendar los ítems que el usuario probablemente prefiera, basándose en su perfil, el perfil de otros usuarios, los datos del proveedor de las posibles preferencias, entre otros criterios a tomar en cuenta; estas recomendaciones son utilizadas para

seleccionar ítems del universo de alternativas. En la práctica, a la hora de concebir un sistema recomendador, muchas veces no se instancian algunas partes de este modelo, todo depende de los requerimientos y objetivos a cumplir.

En la concepción y diseño de un sistema recomendador suelen tenerse en cuenta cuatro aspectos fundamentales. Estos son: la forma de representar las preferencias (14), la manipulación de los roles dentro del sistema y la comunicación usuario-usuario (15), los algoritmos para manipular las recomendaciones (16) y la forma de interacción usuario-máquina a la hora de representar las recomendaciones (17).

1.2 Estructura de un Sistema de Recomendación

El usuario solicita la recomendación y el sistema, a partir de la información que tiene almacenada, realiza el proceso de recomendación y brinda los resultados, ver en la **Figura 2**. Es por esto que la estructura de estos sistemas está dada por:



Figura 2. Estructura de un Sistema de Recomendación. Elaboración propia

1. Las entradas y salidas del proceso de generación de la recomendación: la entrada es la información que el sistema ingresa del usuario y las salidas son las recomendaciones finales. (18)

2. El método usado para generar las recomendaciones (Proceso de Recomendación): es la manera en que el proceso de recomendación se va a desenvolver. Existen tres métodos generales que pueden ser utilizados ya sea uno o varios de ellos en un mismo SR. (18)

- I. Recuperación pura o recomendación nula: es, de forma general, un sistema de búsqueda que permite al usuario examinar ítems en una Base de Datos.
- II. Recomendaciones seleccionadas manualmente por expertos: las recomendaciones son seleccionadas por expertos en el tema a partir de sus propias preferencias, hacen comentarios

para una mayor información y son puestas en una lista disponible para todos los usuarios del sistema.

- III. Resúmenes estadísticos calculados en función de las opiniones del conjunto de usuarios: se hacen cálculos concretos sobre los usuarios que prefieren a un determinado ítem y a partir de esto es que se hacen las recomendaciones. Estos métodos son muy generales y simples por lo que no se consideran totalmente métodos para generar recomendaciones. Existen otras posibilidades más específicas, las cuales se evidencian en los dos grandes grupos de Sistemas de Recomendación: los basados en contenido y los de filtrado colaborativo.

Según el estudio anterior, se decide utilizar como método para generar las recomendaciones los resúmenes estadísticos, ya que las sugerencias se van a obtener a partir de valoraciones que hace el usuario sobre un ítem en específico.

3. El grado de personalización: según el grado de personalización los Sistemas de Recomendación también se clasifican en (18):

- I. No personalizados: brindan las mismas recomendaciones a todos los usuarios.
- II. Personalización efímera: las recomendaciones son dadas según el comportamiento y acciones del usuario en su sesión actual de navegación.
- III. Personalización persistente: estos sistemas están basados en el perfil de los usuarios, por lo que hacen uso de métodos de filtrado colaborativo, filtrado basado en contenidos o correlaciones entre ítems. Brindan recomendaciones con mayor grado de personalización.

Una vez estudiado el grado de personalización, se define que la herramienta a desarrollar utilizará la personalización efímera, ya que se toma en cuenta las interacciones que realiza el usuario en la plataforma para obtener las recomendaciones.

1.3 Clasificación de los sistemas de recomendación

A partir de la bibliografía consultada y al considerar los aspectos establecidos anteriormente, los sistemas de recomendación pueden recibir diferentes clasificaciones en dependencia del tipo de información que utilizan para realizar sus recomendaciones. A continuación, se expone la clasificación más aceptada y reconocida por materiales de referencia. (9) **Figura 3:**



Figura 3. Clasificación de los Sistemas de Recomendación. Elaboración propia

1.3.1 Sistemas basados en contenido

Los sistemas de recomendación basados en contenido intentan recomendar elementos que son similares a aquellos que el usuario previamente valoró en el pasado. Se enfocan en algoritmos de aprendizaje de preferencias de usuarios y filtran nuevos elementos, teniendo en cuenta los que más satisfacen las preferencias de estos. Entre las ventajas que posee este tipo de SR se puede destacar que el sistema puede generar recomendaciones sin la necesidad de contar con un historial previo, por lo que permite realizar predicciones independientemente del historial del usuario. Como principal desventaja se tiene la sobre-especialización, donde el usuario está limitado a que le recomienden ítems similares a los que recomendó. Por tanto, las recomendaciones suelen ser siempre muy similares dado que se basan en la misma información. (19)

Los sistemas basados en contenido son aquellos que recomiendan los ítems a los usuarios basándose exclusivamente en la descripción del ítem y de un perfil de los intereses del propio usuario (20). Estos tienen en cuenta la filosofía de “muéstrame las cosas que he seleccionado, o que me han gustado anteriormente”, aprenden a través de la retroalimentación del individuo con el sistema. Esta retroalimentación puede hacerse de manera explícita, preguntándole directamente al usuario su criterio acerca de determinado ítem, o de manera indirecta, al considerar criterios tales como los sitios frecuentados por el usuario, el tiempo que se demora dentro de una página en específico, y otros que de cierta manera puedan también informar de igual forma las preferencias.

Este tipo de sistemas suele ser utilizado en una gran variedad de dominios que oscilan entre la recomendación de páginas web, artículos noticiosos, restaurantes, programas de televisión, y artículos para vender. A pesar de diferir cada una de estas aplicaciones específicas, todas tienen en común el hecho de contener un modelo que describe los ítems que pueden ser recomendados, un modo de crear

un perfil de usuario que describirá los tipos de ítems que él prefiere y un modo de comparar cada uno de los ítems a recomendar con el perfil del usuario, para determinar cuáles de ellos serán recomendados.

1.3.2 Sistemas basados en filtrado colaborativo

El filtrado colaborativo es otra de las formas de filtrado de información que permite recomendar los elementos que son preferidos por usuarios similares o predecir la utilidad de ciertos elementos para un usuario particular. Las preferencias de los usuarios utilizadas en el sistema son representadas por una evaluación numérica, las cuales pueden obtenerse a partir de diferentes acciones que realice el usuario. (21)

El filtrado colaborativo abarca técnicas orientadas a identificar personas con similares intereses para, basándose en esto, posteriormente construir las recomendaciones. La exitosa implementación de este enfoque se basa en tres condicionales fundamentales que son: la amplia participación de personas dentro del sistema, facilita encontrar otras con intereses similares; la existencia de una forma sencilla de representar los intereses dentro del sistema; y la existencia de algoritmos capaces de relacionar de forma eficiente las personas correspondientes.

Específicamente, mediante el filtrado colaborativo los usuarios expresan sus preferencias a través de la ponderación de ítems que se les presenta, esto sirve para crear un perfil aproximado del usuario. Una vez creado cada uno de los perfiles, cuando un nuevo usuario solicita recomendaciones, el sistema asocia los “ratings” suministrados por este contra los “ratings” que conforman el perfil de los usuarios anteriores, construyendo una lista de usuarios semejantes, más conocida como usuarios más cercanos. Al combinar esta lista, se devuelve un listado de recomendaciones con los ítems mejor evaluados por estos usuarios y que no han sido aún evaluados por el nuevo individuo.

Entre las principales fortalezas de este enfoque está la posibilidad de una total personalización de las recomendaciones y la simplicidad de la representación desde el punto de vista computacional. Unido a esto, como fortaleza también es de remarcar la existencia de una uniformidad en cuanto a la instanciación de los roles. A diferencia de otras filosofías, en el filtrado colaborativo el constructor de recomendaciones y el consumidor de estas está representado por la misma persona, puesto que mientras más recomendaciones o evaluaciones realice el usuario, más exacta será la representación de su perfil y por tanto se incrementará la exactitud de las recomendaciones que este reciba de parte del sistema.

El filtrado colaborativo es considerado una de las tecnologías más potentes de personalización dentro del amplio concepto de la web adaptativa. Entre las aplicaciones pioneras y más representativas de esta técnica se encuentra Amazon (22), importante portal de ventas por internet, y GroupLens, la que

constituye el principal referente arquitectónico y algorítmico dentro de este tipo de sistemas de recomendación. A continuación, se especifican los tipos de filtrado colaborativo:

- Filtrado colaborativo basado en usuario (user-based): sugiere a cada usuario aquellos productos que han interesado a sus vecinos. Se considera que dos usuarios son vecinos, si los mismos presentan preferencias similares, han clasificados los mismos productos en sus perfiles y les han asignado a estos índices parecidos.
- Filtrado colaborativo basado en producto (ítem-based): un producto es recomendado a un usuario si el mismo es similar a los incluidos en su perfil. En este caso, se considera que dos productos son similares si los usuarios que han valorado uno de ellos tienden a valorar el otro asignándole índices de interés parecidos.

En ambas técnicas, el objetivo es predecir el nivel de interés del usuario en relación a un producto dado y, en función de esto, sugerirlo (o no) a dicho usuario. En este sentido, Iniesta Almudena Ruiz identifica dos fases para este proceso (23):

1. En los enfoques colaborativos basado en usuario, la primera fase selecciona a los usuarios cuyas preferencias son similares a los del usuario activo. Por el contrario, los enfoques basados en producto extraen las preferencias del usuario que son más similares al producto objetivo. Luego se forma el vecindario, ya sea de los usuarios o de los productos definidos en su perfil, que estará formado por los usuarios y productos más relevantes.
2. En la segunda fase, el sistema debe predecir el nivel de interés del usuario en relación del producto objetivo. Para ello, los enfoques basados en usuario consideran el nivel de interés de los vecinos del usuario activo en relación al producto objetivo. Por el contrario, la versión colaborativa basada en producto considera el nivel de interés del usuario en relación a aquellos productos de su perfil que son más similares al objetivo.

1.3.3 Sistemas basados en datos demográficos

Los sistemas basados en datos demográficos son similares a los colaborativos, solo que en el cálculo de los usuarios más cercanos usan información demográfica, es decir, elementos basados en el perfil demográfico de los usuarios como la edad y el sexo. Por ejemplo, se les recomienda información a los usuarios de determinados sitios web basados en su idioma o país. Dichas sugerencias pueden ser personalizadas de acuerdo a la edad del usuario (21).

Los beneficios que posee este sistema frente a los sistemas basados en filtrado colaborativo es que no requieren un largo historial de valoraciones para hacer recomendaciones a un usuario. Además, los

sistemas basados en datos demográficos clasifican a los usuarios en grupos y hace recomendaciones de acuerdo con el grupo (21).

1.3.4 Sistemas híbridos

Los sistemas híbridos utilizan las ventajas de los sistemas anteriores, además de aminorar las desventajas de los mismos con el objetivo de sobrellevar los inconvenientes de estos sistemas para lograr una mayor eficiencia en las recomendaciones y hacer llegar al usuario las mejores de ellas.

Como los híbridos están basados en los tres tipos de Sistemas de recomendación ya mencionados, se hace una mezcla entre los funcionamientos de estos, por lo cual se crea el perfil de cada usuario a partir de sus predilecciones anteriores y después se comparan estos perfiles para encontrar las similitudes entre usuarios.

Es decir, los resultados se obtienen a partir de las preferencias del usuario y a su vez de las de los usuarios con preferencias parecidas a las de él. (24)

Los enfoques básicos de la recomendación, anteriormente expuestos, han sido utilizados de manera exitosa en disímiles campos y escenarios. Pese a su difusión, de forma general cada uno de ellos presenta varias desventajas relacionadas principalmente con la necesidad de un gran número de usuarios adheridos al sistema para un correcto funcionamiento del mismo y con la tendencia a una sobre-especialización que aumenta a medida que crece la utilización de la aplicación por parte del usuario.

Con vista a mitigar estas desventajas, una opción muy utilizada es la de combinar la recomendación basada en contenido con el filtrado colaborativo, lo que conforma un sistema de recomendación con características híbridas. A través de estos, la construcción de preferencias se basa tanto en los intereses del usuario, como en los intereses de los usuarios similares, de esta forma se coincide con dos de las técnicas específicas antes expuestas.

Varios autores han demostrado cómo a través de este enfoque la calidad de las recomendaciones aumenta considerablemente (25), constituyendo de esta forma una alternativa a tener en cuenta a la hora de diseñar un sistema de este tipo.

Fundamentación de la elección

Para la implementación de la solución y tras analizar las características de cada tipo de recomendador, se pretende realizar un SR basado en el filtrado colaborativo, ya que, según sus características, es ideal para sistemas donde exista una amplia comunidad de usuarios que colaboren en las recomendaciones a un determinado usuario de esa misma comunidad. Dicha característica está presente en la Plataforma Educativa ZERA, la cual cuenta con una comunidad de más de 2000 usuarios, donde se podrá

implementar este tipo de SR con éxito. Además, los SR con filtrado colaborativo brindan las recomendaciones a los usuarios basándose en la idea de que a un usuario le gustará un elemento si a otros usuarios con preferencias similares le gustó. También diversificará las opciones de cursos y contenidos sugeridos al usuario.

Para el caso en que los usuarios cuenten con un bajo nivel de interactividad con los cursos y contenidos publicados en la Plataforma Educativa ZERA, un sistema basado en filtrado colaborativo constituiría una solución acertada, ya que realizaría las recomendaciones a partir de los datos proporcionados por los otros usuarios de la comunidad. Bajo condiciones de poca actividad en la plataforma, un SR basado en contenidos brindaría sugerencias acotadas a un pequeño espacio de alternativas. Implementar un SR basado en contenidos constituiría una opción apropiada para la plataforma, solamente como parte de un sistema híbrido.

Se descartó utilizar un SR híbrido, ya que posee una alta complejidad y el módulo a desarrollar se encuentra en su primera versión; sin embargo, se pretende implementarlo en futuras versiones.

1.4 Clasificación de los algoritmos para el filtrado colaborativo

En el epígrafe anterior se describen los cuatro tipos en los que se clasifican los sistemas de recomendación: filtrado basado en contenido, demográfico, híbrido y colaborativo. Este último fue elegido para el desarrollo de la solución.

Dentro de los sistemas de recomendación basados en filtrado colaborativo, existen dos clasificaciones para los algoritmos, los que son basados en memoria y los basados en modelos como se muestra en la siguiente figura (26):

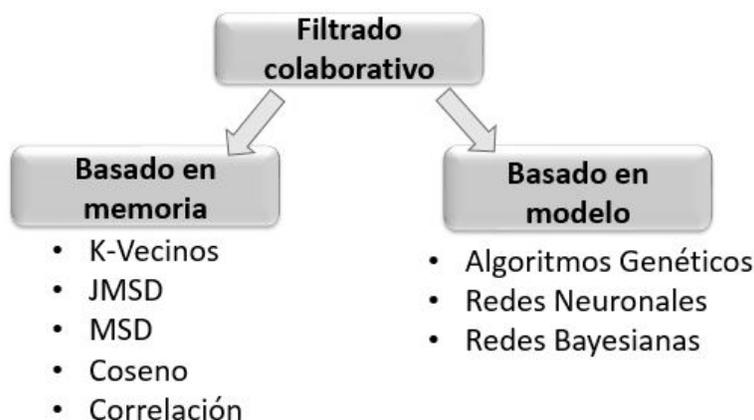


Figura 4. Clasificación de los algoritmos para el filtrado colaborativo. Elaboración propia

Métodos basados en memoria

Los métodos basados en memoria emplean métricas de similitud para determinar el parecido entre una pareja de usuarios. Para ello calculan los ítems (cursos) que han sido valorados por ambos usuarios y comparan dichos votos para calcular la similitud. (26)

Métodos basados en modelos

Los métodos basados en modelos utilizan la matriz de votaciones para crear un modelo a través del cual establecer el conjunto de usuarios similares al usuario activo. Algunos ejemplos de estos modelos son los clasificadores bayesianos, las redes neuronales y los algoritmos genéticos. (26)

Para la implementación del módulo recomendador de cursos y contenidos en la Plataforma Educativa ZERA, se utilizará uno de los métodos basados en memoria, en particular el de los K-vecinos más cercanos (KNN, por sus siglas en inglés de K-Nearest Neighbours); como métrica de similitud se emplea la distancia euclidiana para determinar el parecido entre una pareja de usuarios.

Se decide utilizar el algoritmo KNN ya que, según el estudio realizado, es uno de los más empleados para desarrollar sistemas de recomendación; algunos autores como (26) afirman que el algoritmo KNN es la mejor técnica para realizar recomendaciones. Es fácil de implementar en un lenguaje de programación y arroja, sobre un mismo conjunto de datos, resultados similares a los obtenidos con otras técnicas más complejas, asegura (27). Además, para evaluar cualquier técnica, se tienen en cuenta 3 aspectos importantes: facilidad para interpretar la salida, tiempo de cálculo y poder predictivo; donde KNN es comúnmente usado por su alta facilidad de interpretación y su bajo tiempo de cálculo al compararlo con otros algoritmos de clasificación como se muestra en la siguiente tabla (27):

Tabla 1. Comparación del KNN con otros algoritmos.

Criterio/Algoritmo	Logistic Regression	CART	Random Forest	KNN
1. Facilidad para interpretar la salida.	2	3	1	3
2. Tiempo de cálculo.	3	2	1	3
3. Poder predictivo.	2	2	3	2

K-Vecinos más Cercanos

K-Vecinos más Cercanos es uno de los algoritmos de clasificación más básicos pero esenciales en el Aprendizaje Automático (ML, por sus siglas en inglés de Machine Learning). Pertenece al dominio de aprendizaje supervisado y encuentra una aplicación intensa en el reconocimiento de patrones, la extracción de datos y la detección de intrusos. KNN se ha utilizado en la estimación estadística, el

reconocimiento de patrones y a principios de la década del 70 como una técnica no paramétrica (28). A continuación, se presentan los pasos del algoritmo:

1. Cargar los datos de las valoraciones de los usuarios a los cursos.
2. Elegir el valor de K .

La mejor manera de elegir el valor óptimo para K es inspeccionar los datos. En general, un valor K grande es más preciso, ya que reduce el ruido general, pero no hay garantía. Históricamente, el K óptimo para la mayoría de los conjuntos de datos ha sido entre 3-10. Eso produce resultados mucho mejores que para el caso 1NN.

3. Hallar la distancia euclidiana entre el usuario activo y el resto de usuarios de la base de datos.
4. Ordenar de menor a mayor las distancias obtenidas.
5. Tomar los K primeros valores del arreglo ordenado (vecinos más cercanos). (27)

Se basa en recomendar a un usuario (en este caso usuario activo) los ítems (películas, libros, viajes; en este caso, cursos y contenidos) que les han gustado a usuarios con gustos similares al usuario activo, de ahí el nombre de los k -vecinos, donde los vecinos son usuarios con gustos similares al usuario activo. Por ejemplo, si a un usuario $U1$ le ha gustado las películas El Padrino, Terminator y Titanic y al usuario activo (Ua) le ha gustado también El Padrino y Terminator, es muy probable que, al ser usuarios con gustos muy similares, a Ua también le guste la película Titanic, por tanto, el sistema de recomendación deberá recomendar esta película a Ua . (26)

Lo mismo sucede para un usuario en la Plataforma Educativa ZERA, si un usuario $U1$ le ha gustado (ha valorado) los cursos Matemática, Física y Álgebra Lineal y al usuario activo (Ua) le ha gustado también los cursos Física y Álgebra Lineal, se puede decir que $U1$ y Ua son vecinos cercanos, tienen gustos muy similares, por lo que, es muy probable que a Ua también le guste el curso Matemática, por tanto, el sistema debe recomendárselo.

Obtención del vecindario

La Plataforma Educativa ZERA cuenta con alrededor de 80 cursos con sus respectivos contenidos y, a su vez, también tiene miles de usuarios registrados a los que debe recomendar. Para recomendar a un usuario (activo) con la técnica de los k -vecinos, el usuario previamente debe valorar una serie de cursos para que se sepan cuáles son sus gustos. A su vez, los otros cientos de usuarios deben de haber valorado también un conjunto de cursos. De esta forma, se tiene lo que se denomina la "Matriz de votos", **Ver Tabla 2** que representa los votos (en una escala de 0-5) que han emitido los usuarios sobre los cursos, el valor 0 indica que el usuario no valoró ese curso.

Con la matriz de votos, se puede calcular lo que se denomina "Similitud entre usuarios", que no es más que un valor que indica el grado de cercanía que existe entre usuarios. La similitud entre usuarios se calcula con la métrica de la distancia euclidiana, la cual determina "las distancias" que hay entre cada par de usuarios.

Se decide utilizar esta métrica en vez de otras como la distancia Minkowski, correlación de Pearson o correlación del Coseno, ya que es muy sencilla de implementar en un lenguaje de programación; arroja resultados similares a los obtenidos con otras técnicas más complejas sobre un mismo conjunto de datos; y autores como (28) y (27) aseguran que es la principal métrica para el algoritmo KNN.

El sistema le brinda la posibilidad al usuario de evaluar los cursos mediante estrellas según su agrado. A la hora de obtener el vecindario se crea una matriz donde las filas representan los usuarios, las columnas los cursos y los datos muestran las valoraciones hechas por los usuarios a los cursos como se muestra en la siguiente tabla:

Tabla 2. Matriz de votos.

Usuario/ Curso	Matemática	Física	Historia de Cuba	Álgebra Lineal	Base de Datos I	Ingeniería de SW	Curso n
Usuario 1	5	2.5	3	0	1	4.5	2
Usuario 2	4	3.5	2	1	2	3	3
Usuario 3	0	0	2	1	1	1.5	4
Usuario n	0	0	1	1.5	1.5	0.5	5

La obtención del vecindario consiste en buscar un cierto número de usuarios con gustos similares al usuario autenticado, es decir, buscar usuarios que votaron por cursos en los que el usuario autenticado también votó. El vecindario está constituido por los usuarios que tienen cierto nivel de similitud con el autenticado. A continuación, se define la métrica de la distancia euclidiana.

Distancia Euclidiana

La idea de "similar", puede ser pensada como un modo de comparar cosas, si se piensa en tres personas donde dos estudiaron en la misma universidad, tienen casi la misma edad y altura y tienen diferencia de peso de solo un par de kilos, son "casi iguales" con respecto al tercer individuo, que no estudió en la misma universidad, tiene más años, tiene sobrepeso y mide más que los otros dos. Entonces se puede pensar que los dos primeros son "similares" o al compararlos "son más cercanos o parecidos".

La noción de "cercanía" está asociada con la noción de distancia, se puede decir que dos cafeterías tienen una distancia pequeña si las separa una calle, esto si se toma como unidad de medida el número de calles entre ellas. (29)

Un ejemplo puede ser una matriz de datos, donde cada fila representa un individuo y cada columna es una característica la cual puede tomar el valor de -1, 0 o 1. Entonces lo que se hace es realizar el producto de la matriz y su transpuesta para obtener la correlación entre cada uno de los individuos y lo que se usa como métrica es la distancia euclidiana. Con ellos se puede tener una matriz con información sobre las distancias entre los individuos.

La distancia euclidiana o euclídea es la distancia entre dos puntos de un espacio euclídeo, la cual se deduce a partir del teorema de Pitágoras. En general, la distancia euclidiana entre los puntos $X=(x_1,x_2,\dots,x_n)$ y $Y=(y_1,y_2,\dots,y_n)$, del espacio euclídeo n -dimensional, se define como (30):

$$D(X, Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}$$

En un espacio bidimensional, la distancia euclidiana entre dos puntos P_1 y P_2 , de coordenadas cartesianas (x_1, y_1) y (x_2, y_2) respectivamente, es:

$$D(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

En el siguiente ejemplo, se tiene al Usuario 1 ($U1$) como el usuario autenticado y se desea obtener su vecindario para $k=1$. Se han de calcular las distancias entre el $U1$ y el resto de usuarios, para ello se aplica la métrica de similitud de la distancia euclidiana, para comprobar que tan similar es el $U1$ de otros usuarios como se muestra a continuación:

La tabla hace referencia a la matriz de votos **Tabla 2**.

	Usuario/ Curso	Matemática	Física	Historia de Cuba	Álgebra Lineal	Base de Datos I	Ingeniería de SW	Curso n
Usuario autenticado →	Usuario 1	5	2.5	3	0	1	4.5	2
Usuario más cercano →	Usuario 2	4	3.5	2	1	2	3	3
	Usuario 3	0	0	2	1	1	1.5	4
	Usuario n	0	0	1	1.5	1.5	0.5	5

$$D(1,2) = \sqrt{(5-4)^2+(2.5-3.5)^2+(3-2)^2+(0-1)^2+(1-2)^2+(4.5-3)^2+(2-3)^2}$$

$$D(1,2) = 8.25$$

$$D(1,3) = \sqrt{(5-0)^2+(2.5-0)^2+(3-2)^2+(0-1)^2+(1-1)^2+(4.5-1.5)^2+(2-4)^2}$$

$$D(1,3) = 46.25$$

$$D(1,n) = \sqrt{(5-0)^2+(2.5-0)^2+(3-1)^2+(0-1.5)^2+(1-1.5)^2+(4.5-0.5)^2+(2-5)^2}$$

$$D(1,n) = 62.75$$

Se obtiene que el Usuario 2 es el más cercano al Usuario 1, ya que el valor de la distancia euclidiana para ellos es la menor. Para otro valor de k , 2 por ejemplo, se tiene entonces que el vecindario más cercano al Usuario 1 son los Usuarios 2 y 3 en ese orden, ya que son los dos valores de distancia más pequeños.

1.5 Aplicaciones de los sistemas de recomendación

Los sistemas de recomendación han encontrado aplicación en disímiles áreas de la vida cotidiana; es difícil encontrar en estos momentos algún escenario donde la presencia de un Sistema de Recomendación no se traduzca en mejoras para las personas implicadas en este. Dentro de todas las aplicaciones de estos sistemas, es de remarcar su amplia y estratégica utilización en el marco del e-commerce y del e-learning.

E-Commerce y sistemas de recomendación

El comercio electrónico está presente en el competitivo mercado de hoy en día, donde es de vital importancia para las grandes compañías conocer quiénes son sus clientes, cuáles son sus preferencias y cómo evalúan los productos existentes, teniéndose claro que trabajar en base al criterio de los usuarios finales constituye una de las llaves al éxito en muchas empresas con presencia en internet. La manera más efectiva de obtener esta información es a través del estudio detallado del usuario, sin embargo, realizarlo es altamente costoso e implica ejecutar una planificación cuidadosa.

Con el desarrollo de la web, muchos sitios de comercio electrónico han comenzado a retroalimentarse de sus usuarios de manera directa, al permitirle compartir opiniones y evaluar de manera cuantitativa los productos que compran. Esta información recopilada es de un alto valor para la creación de nuevas aplicaciones que perfeccionen la proyección de la empresa o negocio en cuestión, aplicaciones entre las cuales se encuentran sin dudas los sistemas de recomendación. (31)

Uno de los ejemplos más clásicos de la aplicación de los sistemas de recomendación en este campo lo constituye Amazon, sitio de venta de artículos, cuya sección de venta de libros contiene implícitas varias funcionalidades basadas en esta filosofía. Entre estas cabe destacar “Clientes que compraron”, la cual se encuentra en la página de información de cada libro dentro del catálogo y contiene dos listas de recomendaciones, una con los libros frecuentemente comprados por aquellos clientes que compraron el libro en cuestión y otra con los autores leídos por estos mismos clientes.

Como otras aplicaciones populares en internet que implementan sistemas recomendadores pueden citarse (31) a CDNOW, portal concebido para la venta de CDs de música; eBay, considerado como uno de los mayores portales de compra-venta del mundo; Levis, creada con el objetivo de dar soporte a los clientes de la marca y MovieFinder, capaz de localizar filmes con similares características a un filme dado.

Después de estudiar el impacto de los sistemas de recomendación en el comercio electrónico actual, se puede decir que todos estos sitios tienen algo en común, el alto nivel de aceptación por parte de los usuarios de la red global. El estudio demostró que las empresas conocen las preferencias de los usuarios al permitirle compartir criterios entre ellos y valorar los productos que compran, así construyen las recomendaciones personalizadas para cada uno; en caso que el usuario no valore ningún producto, la recomendación se basa en el historial de compra o de los productos más vendidos. La utilización de mecanismos de recomendación en sitios de comercio electrónico trae consigo un incremento en las ventas, lo que influye de forma directa en las ganancias de la empresa; los usuarios se sienten más motivados en el sitio y disminuye el tiempo de búsqueda de cierto producto de interés.

E-Learning y sistemas de recomendación

El e-learning es un sistema de educación electrónico o a distancia en el que se integra el uso de las tecnologías de la información y otros elementos pedagógicos-didácticos para la formación, capacitación y enseñanza de los usuarios o estudiantes en línea. Es ampliamente aceptado que el e-learning puede ser tan rico y tener tanto o más valor que la clase presencial. El diseño instruccional para el e-learning se ha perfeccionado con el transcurso del tiempo y con la acumulación de experiencias. Como consecuencia de esto, actualmente se cuenta con una amplia red de plataformas contenedoras de los más diversos programas educativos, entre ellos puede mencionarse el MIT Open Course Ware con más de 1800 cursos, Online-education con más de 600 cursos, y Learndirect con alrededor de 500 cursos. (32)

Resulta evidente también en esta área, al igual que en el e-commerce, la existencia de un importante desbordamiento de información que imposibilita el acceso efectivo a los recursos disponibles en plataformas como las mencionadas y a otras aplicaciones de semejante corte. La utilización de sistemas recomendadores aplicados al e-learning ha sido una de las soluciones para enfrentar esta problemática, perfeccionan la eficiencia y eficacia del e-learning y contribuyen a incrementar el aprovechamiento por parte de los estudiantes en medio de la situación actual antes descrita (32). Es válido señalar que, aunque la introducción de los sistemas recomendadores en este campo tuvo una adopción tardía con respecto a la aplicación en el e-commerce, estos actualmente revisten la misma influencia e importancia tanto en un campo como en otro.

Pueden citarse varios trabajos que muestran resultados enfocados a implementaciones reales de sistemas recomendadores aplicados al campo de la enseñanza. En (33) se presenta un sistema recomendador colaborativo para cursos de e-learning, el cual permite que profesores de perfil similar compartan los resultados de sus investigaciones, tras aplicar minería de datos de manera local sobre sus propios cursos. Por otra parte, (34) muestra la implementación de un sistema para la recomendación de

artículos científicos y técnicos, capaz de retroalimentarse de la web para la obtención de nueva información basándose en la adaptabilidad del perfil de usuario.

Luego de realizar un estudio de las aplicaciones de los SR en el e-learning, se puede concluir que estos sistemas son muy empleados en sitios de aprendizaje automático como plataformas educativas, debido a los beneficios que aporta en el proceso de enseñanza-aprendizaje. Con su implementación, el usuario encuentra mayor motivación y orientación hacia los cursos o materiales del sistema, facilita el acceso a cursos de interés y le evita perder tiempo en búsquedas generales de cursos a matricular. En todos los casos estudiados, el sistema crea un perfil de usuario donde almacena sus preferencias para poder llevar a cabo las sugerencias. La utilización de mecanismos de recomendación en sitios de aprendizaje automático trae consigo una mejor formación, capacitación y enseñanza de los usuarios en línea en temas de su interés.

1.6 Ejemplos de sistemas de recomendaciones

Desde hace más de 30 años, ha existido una variedad de sistemas informáticos que utilizan mecanismos de recomendación como apoyo al proceso de la toma de decisiones. A continuación, se exponen los de mayor relevancia para la investigación:

Tapestry

Tapestry fue el primer sistema de recomendación basado en el filtro colaborativo, creado en 1992, el cual permitía que los usuarios hicieran anotaciones sobre documentos electrónicos pertenecientes a un grupo de noticias. El enfoque utilizado en este sistema era una mezcla de la técnica de filtrado basado en contenido y colaborativo. Basado en contenido porque permitía que los usuarios creasen filtros a través de sus ítems de interés y colaborativo, pues los usuarios ingresaban las anotaciones con sus opiniones sobre los documentos y de esa forma colaboraban entre ellos para filtrar la información. Tapestry utilizaba el filtrado colaborativo, pero no lo hacía de forma automática. En este caso el sistema no era quien realizaba las sugerencias, sino que permitía a sus usuarios a partir de las anotaciones que se realizaban sobre los artículos, concluir si un determinado documento era de su interés. (35)

PHOAKS

Se trata de un sistema experimental para solucionar el problema de encontrar información relevante y de alta calidad en la web, usa el enfoque colaborativo en el que los usuarios recomiendan determinados ítems a otros usuarios. PHOAKS reconoce, concuerda y redistribuye automáticamente recomendaciones de recursos web extraídos de mensajes de noticias. (36)

Referral Web

Numerosos estudios muestran que una de las formas más efectivas de divulgar información y conocimientos dentro de una determinada organización es a través de una red informal de colaboradores o amigos. Referral Web se basa en la idea de combinar redes sociales con el filtrado colaborativo. (36)

FAB

Sistema orientado a la recomendación de URL que combina el uso de información por extensión con el enfoque colaborativo. (36)

Siteseer

Recomienda páginas web relevantes y usa las listas de favoritos y la organización de registros como una declaración implícita de intereses respecto al contenido subyacente. Además, mide el grado de solapamiento con las recomendaciones a otros usuarios. (36)

GroupLens

El proyecto GroupLens diseña, implementa y evalúa un sistema de filtrado colaborativo para Usenet, un servicio de listas de discusión con un alto volumen de negocio en Internet. (36)

FilmAffinity

FilmAffinity es una página web que se dedica a la recomendación de películas y series, utiliza la técnica de los K-vecinos. Esta página web tiene registradas miles de películas de diversos géneros y, a su vez, también tiene miles de usuarios registrados a los que debe recomendar. Para el sistema poder recomendar, el usuario previamente debe valorar un conjunto de películas para que se sepan cuáles son sus gustos. (26)

Tutellus

Tutellus es una plataforma educativa independiente, ya que no depende de universidades ni se asocia a ellas. Ofrece cientos de cursos online pero los que más éxito tienen son los relacionados con el emprendimiento, la tecnología y los negocios. Al igual que otras plataformas educativas como Miríada X, Coursera y Floqq, Tutellus también recomienda cursos a sus usuarios. (37)

EDUTEKA

EDUTEKA es un portal educativo, funciona como un repositorio, ya que contiene una gran cantidad de recursos de aprendizaje, además, se pueden realizar consultas guiadas, tanto para obtener información reciente sobre diferentes temas educativos, como para recibir sugerencias en la parte didáctica, lo que permite generar aprendizajes más significativos por medio de las actividades, softwares y links a otros sitios. (38)

Según el estudio realizado, gran parte de los sistemas informáticos que utilizan mecanismos de recomendación como apoyo al proceso de la toma de decisiones, emplean el enfoque colaborativo en sus soluciones por encima de otros enfoques como el basado en contenido o el híbrido. Algunos utilizan la técnica de los K-vecinos para obtener sus recomendaciones como es el caso de FilmAffinity. Además, se pudo apreciar que, en todos los casos estudiados, recomiendan al usuario contenido propio; por ejemplo, la plataforma Coursera, recomienda cursos almacenados en su sistema y no de terceros.

1.7 Metodología de desarrollo adoptada

Las metodologías son utilizadas para solucionar problemas existentes en la producción de software. Mediante la utilización de las mismas se realizan procedimientos, técnicas, se genera documentación y se utilizan herramientas con el objetivo de lograr un mejor producto software. Esta no es más que la definición de un conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto a modo de plantilla que explica los pasos necesarios para la finalización del mismo. (39)

Cada metodología se diferencia en la forma que gestionan con precisión los artefactos, roles y actividades del proyecto. Pero de manera general se puede encontrar en dos grandes grupos: metodologías tradicionales (o también denominadas Metodologías Pesadas, o Peso Pesado) y metodologías ágiles.

Las tradicionales establecen durante todo el proceso de desarrollo un mayor énfasis en la planificación y control del proyecto, y en la especificación precisa de los requisitos y el modelado. Por otro lado, las ágiles están más orientadas a la generación de código con ciclos muy cortos de desarrollo, enfocándose a equipos de desarrollo pequeños, por lo que hace especial hincapié en aspectos humanos asociados al trabajo en equipo e involucra activamente al cliente en el proceso. (40)

Un resumen más sencillo de la comparación que se establece entre las dos metodologías antes mencionadas, se muestra en la siguiente tabla: Comparación de metodologías, tomado del artículo Metodologías tradicionales VS. Metodologías ágiles. (41)

Tabla 3. Diferencias entre Metodologías Ágiles y Tradicionales.

Metodologías Tradicionales	Metodologías Ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.	Basadas en heurísticas provenientes de prácticas de producción de código.
Cierta resistencia a los cambios.	Especialmente preparados para cambios durante el proyecto.
Impuestas externamente.	Impuestas internamente (por el equipo).

Proceso mucho más controlado, con numerosas políticas/normas.	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.
Más artefactos.	Pocos artefactos.
Más roles.	Pocos roles.
Grupos grandes y posiblemente distribuidos.	Grupos pequeños (10 integrantes) que trabajan en el mismo sitio.
La arquitectura del software es esencial y se expresa mediante modelos.	Menos énfasis en la arquitectura del software.
Existe un contrato prefijado.	No existe contrato tradicional o al menos es bastante flexible.

Después de realizar el estudio sobre las metodologías de desarrollo de software, se cuenta con la necesidad de adaptarse a las exigencias y condiciones del cliente, en cuanto al ahorro de tiempo y costos en el ciclo de desarrollo y entrega de la solución con calidad. Para guiar el desarrollo de la solución, se utiliza la metodología AUP-UCI.

AUP-UCI

El Proceso Unificado Ágil (AUP, por sus siglas en inglés) es una versión simplificada del Proceso Racional Unificado (RUP, por sus iniciales en inglés). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio, hacen uso de técnicas ágiles y conceptos que aún se mantienen válidos en RUP.

Producto a la no existencia de una metodología de software universal 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 UCI. Dentro de las técnicas ágiles que implementa esta nueva variación y que se mantienen válidas en su antecesor, se pueden mencionar: (42)

- ✓ Desarrollo dirigido por pruebas.
- ✓ Modelado ágil.
- ✓ Gestión de cambios ágil.
- ✓ Refactorización de base de datos.

El uso de la técnica de modelado ágil permite encapsular los requisitos funcionales en Historias de Usuario (HU) o descripción de requisitos por procesos.

A continuación, se presentan las fases propuestas por dicha metodología durante el proceso de creación de software.

Tabla 4. Fases de la metodología AUP-UCI.

Fases	Objetivos
Inicio	<ul style="list-style-type: none"> ✓ Planeación del proyecto. ✓ Estudio inicial de la organización cliente. ✓ Alcance del proyecto. ✓ Estimaciones de tiempo, esfuerzo y costo.
Ejecución	<ul style="list-style-type: none"> ✓ Ejecución de las actividades requeridas para desarrollar el software. ✓ Ajuste de los planes del proyecto. ✓ Modelado del negocio. ✓ Levantamiento de requisitos. ✓ Elaboración la arquitectura y el diseño. ✓ Implementación y liberación del producto.
Cierre	<ul style="list-style-type: none"> ✓ Análisis de los resultados y ejecución del proyecto. ✓ Cierre del proyecto.

Se decide utilizar la metodología AUP-UCI, encapsulando los requisitos a través del escenario 4, ya que es la metodología recomendada en la institución para guiar los procesos de desarrollo de software, responde con facilidad a los cambios continuos y cumple con las políticas de desarrollo de la universidad. Además, es la metodología empleada en el proyecto Plataforma Educativa ZERA, por lo tanto, se trabajará acorde al proyecto y se generarán los artefactos necesarios para futuras mejoras del módulo a desarrollar.

1.8 Herramientas y tecnologías

Para el desarrollo de la aplicación se hace necesario el uso de herramientas y tecnologías. Se realiza un estudio para valorar cuáles serían las adecuadas teniendo en cuenta la naturaleza del problema que se mueve dentro del contexto académico y científico. Se hace énfasis en las principales herramientas y tecnologías definidas como parte de la arquitectura de desarrollo de software de la Plataforma Educativa ZERA.

1.8.1 Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés de Integrated Development Environment) es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien poder utilizarse para varios. Puede denominarse como un entorno de programación que ha sido tratado como un programa aplicación. Esto significa que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. (43)

En el caso de la solución, se decide utilizar el IDE NetBeans, ya que es una aplicación gratuita y sin restricciones de uso; no se puede decir lo mismo para otros IDEs analizados y descartados como PhpStorm y WebStorm que son privativos de la empresa JetBrains y su licencia es de pago. NetBeans es multiplataforma y soporta lenguajes dinámicos como PHP y JavaScript. Además, permite la integración con el marco de trabajo Symfony y con la consola de PHP, por lo que permite ejecutar comandos internos de dicho lenguaje.

NetBeans 8.1

NetBeans es un IDE modular y escrito en el lenguaje de programación Java. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general (framework) para compilar cualquier tipo de aplicación. NetBeans soporta varios lenguajes de programación como Java, C/C ++, XML, HTML, PHP y JavaScript. Es multiplataforma, es decir, puede instalarse en varios sistemas operativos como Windows, Linux o Mac OS. (44)

Ofrece diferentes vistas de los datos, desde varias ventanas de proyectos a herramientas útiles para la creación de aplicaciones y su gestión de manera eficiente, lo que le permite desglosar sus datos de forma rápida y sencilla. También proporciona plantillas de código, consejos de codificación y herramientas de refactorización. Además, permite la integración con el marco de trabajo Symfony 2, el cual posee algunas bibliotecas y plugins que el programador puede agregarle a su aplicación. También posee integración con la consola de PHP, por lo que se puede ejecutar comandos internos de dicho lenguaje.

Algunas características de NetBeans son: (44)

- Editor de código rápido e inteligente.
- Gestor de proyectos fácil y eficiente.
- Desarrollo rápido de la interfaz de usuario.
- Disponible en muchos idiomas.
- Posee una amplia documentación y una gran comunidad de usuarios.

1.8.2 Frameworks de desarrollo

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, permite al desarrollador crear código más legible y fácil de mantener. Por último, un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Symfony 2.7.16

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Automatiza las tareas más comunes, lo que permite al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web. (45)

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas Unix como en plataformas Windows. A continuación, se muestran algunas de sus características:

- Independiente del sistema gestor de bases de datos.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.
- Los formularios incluyen validación automatizada y relleno automático de datos (“repopulation”), lo que asegura la obtención de datos correctos y mejora la experiencia de usuario.
- La gestión de la caché reduce el ancho de banda utilizado y la carga del servidor.
- El soporte de e-mail incluido y la gestión de APIs permiten a las aplicaciones web interactuar más allá de los navegadores. (45)

El framework de desarrollo Symfony brinda muchas posibilidades para los programadores que lo deseen usar, debido a su simplicidad y robustez. Se decide escoger Symfony dado que es un framework acorde a la arquitectura empleada en el desarrollo de la solución, facilita el desarrollo de aplicaciones web al evitar hacer actividades que ya están hechas. Reduce el tiempo de creación de una aplicación, ya que viene con componentes que se encargan de administrar las rutas de manera fácil. Trae incorporadas funcionalidades como migraciones de bases de datos, validación de formularios, administración de

sesiones, autorización y autenticación, procesadores de correo electrónico, barra de depuración e interfaz gráfica para analizar el desempeño de la aplicación.

JQuery 1.11.2

jQuery es un framework para el lenguaje Javascript que implementa una serie de clases (de programación orientada a objetos) que permite programar sin tener en cuenta el navegador que tiene el usuario. Ofrece una infraestructura con la que se tendrá mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. (46)

Para el desarrollo del módulo se va a utilizar JQuery ya que permite agregar plugins de forma fácil, lo que implica un ahorro substancial de tiempo y esfuerzo. Además, la licencia cuenta con soporte constante y rápido. Por otro lado, proporciona el manejo rápido de propiedades y CSS que se emplearán en la solución propuesta.

Bootstrap 3.3.2

Bootstrap es un excelente framework que permite crear interfaces de usuario limpias y totalmente adaptables a todo tipo de dispositivos y pantallas basado en HTML, CSS y JavaScript (47). Esta herramienta ayudará a agilizar la creación de la interfaz de nuestra página web y permitirá crear un diseño limpio, intuitivo, usable y de poco peso, por lo que la carga de nuestra web será muy rápida. Es compatible con la mayoría de navegadores web y ofrece las herramientas necesarias para crear cualquier tipo de sitio web con sus propios estilos y librerías. (48)

Para el desarrollo del módulo se va a utilizar Bootstrap, ya que simplifica el proceso de maquetación, sirve de guía para aplicar las buenas prácticas y los diferentes estándares. Por otro lado, se puede hacer uso de elementos web como iconos y desplegables, combina los lenguajes HTML5, CSS y JavaScript. Es de código abierto, por lo que se puede usar de forma gratuita y sin restricciones.

1.8.3 Herramienta CASE

Tras estudiar y valorar un trío de herramientas de Ingeniería de Software Asistida por Computadora (CASE, por sus siglas en inglés de Computer Aided Software Engineering) más importantes, dígase Visual Paradigm, Rational Rose y MagicDraw, se resolvió que Rational Rose no es aplicable en este caso, ya que obliga a desarrollar en el sistema operativo Windows por lo que no es multiplataforma; MagicDraw por su parte, solo permite el desarrollo de código para un reducido grupo de lenguajes como C#, C++ y Java.

Sin embargo, aunque en un principio no se descartó usar MagicDraw, la tercera opción, Visual Paradigm en su versión 8.0, es la escogida finalmente por ser la que más información se puede encontrar y en la que más experiencia tiene el desarrollador, dado que es objeto de estudio por parte de la universidad; es

más usable que otras herramientas, resulta más cómodo trabajar con ella, ya que permite generar diversos prototipos como el de la base de datos y así ahorrar tiempo. Además, está disponible en varias plataformas y permite crear diagramas durante todo el ciclo de vida de desarrollo de un proyecto de manera fácil, entendible y rápida.

Visual Paradigm 8.0

Visual Paradigm for UML (VP) es una herramienta CASE que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite construir diagramas de diversos tipos, código inverso, generar código desde diagramas y generar documentación. (49)

Entre sus características fundamentales se tienen:

1. Multiplataforma: soportada en plataforma Java para Sistemas Operativos Windows, Linux, Mac OS.
2. Interoperabilidad: intercambia diagramas y modelos con otras herramientas. Soporta la importación y exportación a formatos XMI y XML y archivos Excel. Permite importar proyectos de Rational Rose y la integración con Microsoft Office Vision.
3. Ingeniería de código: permite la generación de código e ingeniería inversa para los lenguajes: Java, C, C++, PHP, XML, Python, C#, VB .Net, Flash, ActionScript, Delphi y Perl.
4. Integración con entornos de desarrollo: apoyo al ciclo de vida completo de desarrollo de software en IDEs como: Eclipse, Microsoft VisualStudio, NetBeans, Sun ONE, Oracle JDeveloper, Jbuilder y otros.
5. Brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.

1.8.4 Servidor Web

NGINX es un servidor HTTP libre, de código abierto, de alto rendimiento y proxy inverso, así como un servidor proxy IMAP o POP3. Es conocido por su alto rendimiento, estabilidad, conjunto de características ricas, configuración sencilla y bajo consumo de recursos. No se basa en hilos para manejar las solicitudes, en su lugar, utiliza una arquitectura asíncrona basada en eventos mucho más escalable. Esta arquitectura utiliza cantidades pequeñas, pero más importantes, predecibles de memoria bajo carga. Incluso si no espera manejar miles de solicitudes simultáneas, todavía puede beneficiarse del alto rendimiento y de la pequeña huella de memoria de NGINX. Además, escala en todas las direcciones: desde el más pequeño VPS hasta grandes grupos de servidores. (50)

Para el desarrollo del módulo se utiliza el NGINX en su versión 1.10.3, ya que utiliza módulos externos que se pueden agregar al servidor, lo cual lo hace mucho más liviano y ágil. Además, es multiplataforma,

fácil de instalar y compatible con una gran cantidad de Sistemas de Gestión de Contenido (CMS, por sus siglas en inglés de Content Management System) existentes en el mercado como WordPress, Drupal y Joomla.

1.9 Lenguajes de programación

La aplicación a desarrollar es, por definición, una aplicación web. En esta sección se especifican las características más importantes de tres de los lenguajes de programación más utilizados para el desarrollo de aplicaciones web, que son Python, Java y PHP, justificándose la utilización de este último para el desarrollo.

Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje multiplataforma. (51)

Usa tipado dinámico y conteo de referencias para la administración de memoria. Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos) (51). Este lenguaje se utiliza para la solución propuesta en conjunto con el framework Django ya que el mismo permite la implementación de algunas funcionalidades del sistema.

Java

Java surgió en 1991 cuando un grupo de ingenieros de Sun Microsystems trató de diseñar un nuevo lenguaje de programación destinado a correr sobre cualquier plataforma: desde computadoras hasta efectos electrodomésticos. Con este fin fueron de los primeros en introducir el concepto de máquina virtual con el objetivo de que sus aplicaciones fueran totalmente independientes del CPU que las procesara.

Sun describe a Java como simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico (52). Se ha desarrollado unido a Internet y es por ello que es ampliamente utilizado en la construcción tanto de sistemas distribuidos, como de complejas aplicaciones de gestión.

El hecho de la dependencia de una máquina virtual, como mismo constituye una ventaja, es también una desventaja. El depender de un intermediario para ejecutar los programas hace a estos más lentos y los vuelve dependientes de la correctitud de la mencionada máquina virtual: en ocasiones el código está bien escrito, pero el programa no trabaja correctamente debido a errores de la máquina virtual.

PHP 7

PHP es un lenguaje de alto nivel específicamente pensado para desarrollar páginas web por lo cual puede ser incrustado en páginas HTML. Es un lenguaje fácil de aprender. Su sintaxis es similar a C, Perl y Java. La versión PHP5 soporta programación orientada a objeto lo cual lo hace mucho más portable, asequible y manejable. Es uno de los lenguajes más usado en el mundo. Dispone de una gran cantidad de interfaces para diferentes sistemas de base de datos además puede conectarse de forma directa a estas (Mysql, Postgresql, Msq, Oracle, dbm, filepro). PHP está disponible para una gran cantidad de sistemas operativos como Unix, Linux, Windows y Free BSD. El código funciona sin tener que modificar los sistemas que lo ejecutan y dispone de acceso, por lo cual se puede agregar otras funcionalidades al lenguaje con total libertad.

PHP, acrónimo recursivo que significa PHP Hypertext Pre-processor, fue creado originalmente por Rasmus Lerdorf en 1994. Es un lenguaje interpretado, diseñado originalmente para la creación de páginas web dinámicas, y con acceso a información almacenada en bases de datos. Fue liberado bajo la PHP License; la Free Software Foundation (FSF) considera como software libre a los productos liberados bajo esta licencia.

Entre sus ventajas más resaltables está el hecho de consumir pocos recursos, por lo que generalmente se ejecuta con rapidez y no tiende a ralentizar el resto de los procesos del sistema operativo (53). Otra importante ventaja es su gran capacidad de conectividad, al utilizar un sistema de extensiones modular hacia interfaces de variado tipo, tales como librerías gráficas, XML, encriptación, etc. Entre las principales desventajas se pueden mencionar que para la realización de proyectos complejos requiere un alto nivel de experiencia, y que por determinadas limitaciones del lenguaje existe un pequeño grupo de tareas que resultan muy difíciles de realizar en el mismo. (54)

Fundamentación de la elección

Para el desarrollo del módulo se utiliza PHP, ya que es un lenguaje multiplataforma, completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos. El código fuente escrito en PHP es invisible al navegador y al cliente, el servidor es el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Posee la capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL. Además, el proyecto ZERA utiliza PHP como lenguaje de programación acompañado de Symfony como framework de desarrollo, lo que posibilita luego de terminar el módulo, una fácil integración con el proyecto.

HTML 5

Para el desarrollo del módulo se utiliza el lenguaje de marcas de hipertexto (HTML, por sus siglas en inglés de HyperText Markup Language), ya que es un lenguaje que utilizan los ordenadores y los programas de diseño, es muy fácil de aprender y escribir por parte de las personas. Se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo. Además, desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica. Se define en la siguiente frase, "HTML es lo que se utiliza para crear todas las páginas web de Internet". (55)

Para el desarrollo del módulo se utiliza HTML en la creación de las interfaces del sistema, donde se aprovechan las peculiaridades del soporte CSS3. HTML es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada, con enlaces que conducen a otros documentos o fuentes de información relacionadas; permite la inserción de contenido multimedia como gráficos, sonido, imagen y video. Además, es multiplataforma, se puede leer o interpretar una página web desde sistemas Windows y Linux.

CSS 3

Las hojas de estilo en cascada (CSS, por sus siglas en inglés de Cascading Style Sheets) son las que ofrecen la posibilidad de definir las reglas y estilos de representación en diferentes dispositivos, ya sean pantallas de equipos de escritorio, portátiles, móviles u otros dispositivos capaces de mostrar contenidos web. Permiten definir de manera eficiente la representación de páginas web y es uno de los conocimientos fundamentales que todo diseñador web debe manejar a la perfección para realizar su trabajo. (56)

Se decide utilizar CSS en el desarrollo del módulo con el fin de añadir estética a las páginas según las pautas de diseño de la interfaz web del sistema. CSS brinda la posibilidad de usarse con otros lenguajes de programación como JavaScript para conseguir efectos dinámicos en las páginas y permite modificar la vista de cada elemento sin cambiar el código HTML, por lo que se ahorra tiempo y esfuerzo de edición.

JavaScript 1.3

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas, incorpora efectos como textos que aparecen y desaparecen, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. (57).

Se utilizará JavaScript en el desarrollo del sistema para incrementar la interacción en la aplicación con respecto a la visualización de los efectos asociados a eventos, lo que permite un manejo de manera más rápida y dinámica. Además, es fácil de integrar y utiliza poca memoria.

1.10 Lenguaje de modelado

Se estudiaron dos lenguajes, UML 2.0 y SysML. Finalmente se decidió no utilizar el segundo ya que está destinado a proporcionar técnicas de modelado a una gran variedad de sistemas sin tener en cuenta la dirección de proyectos de ingeniería de sistemas como aspectos de la planificación, entre otras de vital importancia en el desarrollo del software.

Se escoge como lenguaje de modelado UML, ya que es de fácil comprensión, resulta menos engorroso realizar cambios una vez comenzado a desarrollarse en el mismo, es más sencillo encontrar dificultades o dependencia en los sistemas y permite definir los componentes que se utilizarán para construir el sistema con apoyo de la herramienta de modelado Visual Paradigm.

UML 2.0

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. (58)

UML implementa un lenguaje de modelado común para todos los desarrollos por lo que crea una documentación común que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado en el desarrollo. (59)

1.11 Sistema Gestor de Base de Datos

Los Sistemas de Gestión de Bases de Datos (SGBD) son programas orientados a la administración y diseño de las bases de datos, lo que permite su creación y modificación. Aunque fueron valoradas otras alternativas como Oracle y MySQL, el gestor de base de datos escogido finalmente fue PostgreSQL. La alternativa Oracle fue desechada pese a ser uno de los sistemas de base de datos más complejos dado que este es un producto muy costoso, por lo que es utilizado en grandes empresas y multinacionales de desarrollo de software dado el alto costo que supondría su soporte técnico y mantenimiento. PostgreSQL es la alternativa idónea por ser un SGBD de código abierto y orientado a objetos, cuenta con un entorno amigable y fácil de utilizar, es un sistema que soporta la gran mayoría de las transacciones SQL y trabaja con gran cantidad de lenguajes como Java, Python y PHP.

PostgreSQL 9.5

Para el buen funcionamiento de la aplicación se necesita contar con un SGBD que posibilite la obtención de la información de la forma más sencilla y rápida posible. En la actualidad PostgreSQL es considerado uno de los gestores de base de datos de código abierto más avanzado del mundo, debido a que soporta casi toda la sintaxis SQL.

Para el desarrollo del módulo se utiliza PostgreSQL, ya que es una herramienta libre y multiplataforma. Tiene estabilidad, potencia, robustez, es fácil de administrar y soporta una alta concurrencia de usuarios al mismo tiempo.

PgAdmin 1.22.0

Es una aplicación gráfica para administrar PostgreSQL con licencia de código abierto. Es capaz de gestionar versiones a partir de PostgreSQL 7.3 y superiores ejecutándose en cualquier plataforma. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. Una característica interesante es que, cada vez que se realiza alguna modificación en un objeto, escribe las sentencias SQL correspondientes, lo que hace que, además de una herramienta muy útil, sea a la vez didáctica. También incorpora funcionalidades para diseñar consultas. (60)

Conclusiones parciales

En este capítulo se realizó un estudio del estado del arte de los sistemas de recomendación, se expone de forma breve su basamento teórico y principales características, lo que permite un mejor entendimiento de la investigación. Se analizaron las diferentes clasificaciones a las que pueden ser sometidos, basándose principalmente en la forma de procesar la información. Se llegó a la conclusión de que tienen una amplia aplicación en el mundo de la World Wide Web, comprobándose seguidamente que son también aplicables a las plataformas educativas. Finalmente, se concluyó que la metodología de desarrollo a utilizar será AUP-UCI en su escenario 4, lo que facilita la implementación del módulo gracias a las características que la misma posee. El perfil tecnológico estará conformado por PHP como lenguaje de programación, Symfony como framework de desarrollo, NetBeans como IDE y Visual Paradigm como herramientas CASE.

Capítulo 2. Propuesta de solución

El aspecto más importante de cualquier propuesta de solución es identificar y comprender el problema que el cliente busca resolver. Uno de los puntos del desarrollo de una propuesta de solución es presentar una noción propia del problema, así como la forma de resolverlo, con el fin de convencer al cliente de que tal propuesta es la mejor. En el presente capítulo se procede a realizar la descripción de la propuesta de solución, que incluye el modelo conceptual, los requisitos funcionales y no funcionales, las historias de usuarios y aspectos del análisis como: diagrama de clases del análisis y diagrama de colaboración del análisis. Además, se realizan los diagramas de secuencia, se definen los patrones de diseño y patrón arquitectónico como parte del diseño del módulo.

2.1 Modelo conceptual

Un modelo conceptual o de dominio es una representación visual de las clases conceptuales significativas u objetos del mundo real en un dominio de interés, no de componentes de software. No se trata de un conjunto de diagramas que describen clases de software. Es el artefacto más importante que se crea durante el análisis orientado a objetos. (61)

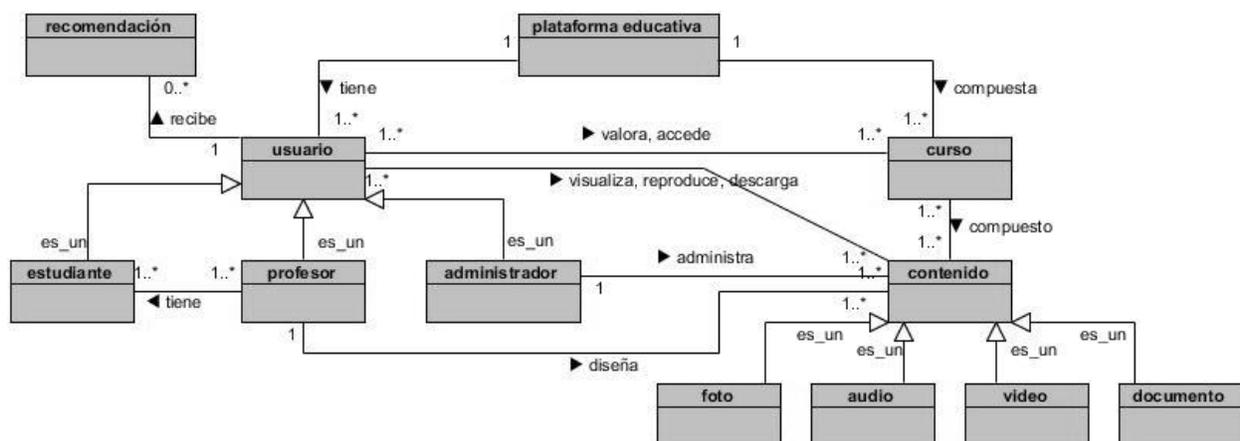


Figura 5. Modelo conceptual

Definición de los conceptos del modelo del dominio

Plataforma educativa: es la Plataforma Educativa ZERA, sistema informático que aumenta el acceso a una educación de calidad para todos los usuarios mediante la gestión de cursos en línea, promueve la enseñanza y el aprendizaje a través de las Tecnologías de la Información y las Comunicaciones.

Usuario: se refiere a los distintos roles que pueden acceder a la Plataforma Educativa ZERA.

Profesor: persona encargada de diseñar los cursos y contenidos en la Plataforma Educativa ZERA.

Administrador: persona encargada de administrar los cursos y contenidos en la Plataforma Educativa ZERA.

Estudiante: persona que recibe los cursos y contenidos diseñados por el profesor en la Plataforma Educativa ZERA.

Recomendación: sugerencia dada al usuario de la Plataforma Educativa ZERA.

Curso: conjunto de contenidos orientados al estudiante en la Plataforma Educativa ZERA.

Contenido: elemento digital diseñado por el profesor, orientado al estudiante de la Plataforma Educativa ZERA con un fin específico en el proceso enseñanza-aprendizaje.

Foto, Audio, Video, Documento: son tipos de contenidos diseñados por el profesor que el usuario puede visualizar, reproducir y descargar en dependencia del tipo que sea.

2.2 Descripción de la propuesta de solución

Para solucionar el objetivo propuesto de desarrollar un módulo de recomendación que facilite a usuarios de la Plataforma Educativa ZERA, el acceso a cursos y contenidos de su interés de acuerdo a su interacción y a la de otros usuarios con el sistema, se propone un módulo que, en primer lugar, sea capaz de guardar en bases de datos la información necesaria sobre la interacción de los usuarios con la plataforma. Es decir, cuando el usuario valore o acceda a un curso, visualice una imagen, cuando reproduzca un audio o un video o cuando descargue un documento, se guarde el usuario que hizo la operación, el tipo de operación (visualizar, acceder, valorar, reproducir, descargar), el tipo de ítem (imagen, video, audio, documento, curso), el identificador del ítem, entre otros elementos.

El usuario tiene la posibilidad de decidir si quiere o no recibir notificaciones de tipo recomendación en pantalla y/o vía correo electrónico. En caso que desee recibir las recomendaciones en pantalla (para el correo electrónico es el mismo procedimiento), se verifica si valoró o no algún curso, en caso afirmativo, se aplica el algoritmo KNN para hallar su vecindario cercano. Si el usuario no ha valorado ningún curso, no se puede obtener su vecindario cercano ya que se desconocen sus preferencias, esta limitante se conoce como arranque en frío y es una desventaja de los sistemas recomendadores basados en filtrado colaborativo; para lidiar con el arranque en frío, se le recomienda al usuario los cursos más populares (los mejores valorados) de la plataforma. Si no desea recibir notificaciones de tipo recomendación, no se realiza el proceso antes descrito.

Después de obtener el vecindario cercano, se procede a encontrar los cursos que hayan sido de mayor interés (mejores valorados) por esos vecinos cercanos y que el usuario activo no se haya matriculado aún en estos cursos para poderse los recomendar. Cabe destacar que, si dos o más vecinos han valorado el

mismo curso, este se escoge solo una vez. La recomendación de contenidos se realiza dentro de un curso, se obtiene el vecindario cercano al usuario y se le recomienda los contenidos de ese curso más vistos (más accedidos) por los vecinos cercanos.

La recomendación (tanto en pantalla como por correo) se realiza cada cierto tiempo. Para el caso de los contenidos, cuando el usuario acceda a un curso y se cumpla el tiempo de la recomendación, se le va a mostrar en pantalla los contenidos de ese curso sugeridos para él. La sugerencia de contenidos no se va a realizar mediante correo electrónico, ya que los correos, en ocasiones, son revisados mucho tiempo después, cuando el usuario ya cerró su sesión en la plataforma, y para los contenidos, se desea que se muestre la sugerencia cuando el usuario interactúe con el curso. En el caso de los cursos, la recomendación se le va a mostrar en pantalla y/o vía correo electrónico. No obstante, si el usuario quisiera recordar la recomendación de cursos antes hecha, puede desplazarse al apartado Notificaciones de la plataforma donde podrá ver lo recomendado.

Para garantizar la seguridad, los usuarios deberán estar previamente autenticados en la plataforma de acuerdo a los roles definidos, con el fin de impedir el acceso no autorizado a las funcionalidades del módulo.

2.3 Especificación de requisitos de software

“La Especificación de Requisitos de Software (ERS) es una de las fases más importante del proceso de desarrollo de software y es en esta fase donde se obtiene la descripción completa del comportamiento del futuro software o producto que se va a desarrollar.” (62)

En la ERS es donde se registran las necesidades del negocio (requisitos del cliente y necesidades del usuario) y se definen los requerimientos que debe cumplir el producto, sistema o software a desarrollar. Por tanto, es el medio de comunicación entre los clientes, usuarios y desarrolladores. (62)

“A menudo, los requerimientos del sistema de software se clasifican como requerimientos funcionales o requerimientos no funcionales.” (63)

Requisitos funcionales

Los requisitos funcionales son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas. En algunos casos, los requisitos funcionales también explican lo que no debe hacer el sistema. (63)

A continuación, se muestran los requisitos funcionales del módulo a desarrollar.

RF1. Interactuar con los cursos y contenidos. El sistema debe almacenar en base de datos la información sobre la interacción de los usuarios con los cursos y contenidos de la Plataforma Educativa ZERA.

Interacción de los usuarios con los cursos: se refiere cuando el usuario accede a un curso.

Interacción de los usuarios con los contenidos: se refiere cuando el usuario visualiza una foto o un documento, reproduce un audio o un video, o cuando descarga un documento.

RF2. Valorar un curso. El usuario puede valorar un curso mediante estrellas de 0-5 con un nivel de aceptación: 1 si considera el curso malo, 2 pobre, 3 regular, 4 bueno y 5 maravilloso.

RF3. Mostrar recomendación de cursos y contenidos obtenidos mediante el algoritmo de la vecindad próxima. El sistema debe mostrar al usuario la recomendación de cursos y contenidos obtenidos mediante el algoritmo KNN.

RF4. Mostrar recomendación con los cursos mejores valorados. El sistema debe mostrar al usuario la recomendación con los cursos mejores valorados de la plataforma.

RF5. Visualizar recomendación de cursos en el apartado Notificaciones de la plataforma. El usuario puede visualizar la recomendación de cursos en la sección Notificaciones de la plataforma.

RF6. Configurar notificaciones de tipo recomendación. El usuario puede configurar si desea o no recibir notificaciones de tipo recomendación en pantalla o vía correo electrónico.

RF7. Notificar recomendación de cursos vía correo electrónico. El sistema debe notificar al usuario vía correo electrónico la recomendación de cursos generada.

Requisitos no funcionales

Los requisitos no funcionales son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones de tiempo, del proceso de desarrollo y de estándares. Los requisitos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema.
(63)

Los requisitos no funcionales identificados en la presente investigación son los siguientes:

Hardware

- Servidor de aplicaciones NGINX: 1.10.3 con memoria RAM: 16 GB, Disco Duro: 500 GB y microprocesador: 6 x 800 GHz.

- Servidor de bases de datos relacional PostgreSQL 9.5 con memoria RAM: 16 GB, Disco Duro: 100 GB y microprocesador: 6 x 800 GHz.

Seguridad

- Declarar los permisos por roles para el acceso al sistema. Validar los datos del cliente, para evitar estados inconsistentes en la información y posibles ataques al sistema. Autenticación basada en las bondades del dominio UCI.

Portabilidad

- El sistema será accesible desde diferentes estaciones de trabajo como pc de escritorio, tablet, laptop y smartphone. Los equipos deben tener un navegador web moderno como Firefox o Chrome.

Software

- La aplicación debe ejecutarse en el sistema operativo Windows XP o superior, o GNU Linux en cualquiera de sus distribuciones.
- Para interactuar con el sistema se debe utilizar el navegador (Mozilla Firefox versión 25 o superior, Google Chrome versión 30 o superior).

Historias de usuario

Las historias de usuario son descripciones, siempre muy cortas y esquemáticas, que resumen la necesidad concreta de un usuario al utilizar un producto o servicio, así como la solución que la satisface. Como muchas otras herramientas ágiles, las historias de usuario surgieron como una respuesta orientada al sector de desarrollo de software, aunque con el paso del tiempo se han utilizado en otros tipos de negocio. Su función principal es identificar problemas percibidos, proponer soluciones y estimar el esfuerzo que requieren implementar las ideas propuestas. (64)

La implantación de este concepto permite añadir una visión más amplia a nuestro proceso de desarrollo. Los siguientes elementos son algunas de las ventajas de trabajar con historias de usuario:

- Son deseos o necesidades muy concretas que se centran en partes definidas del proyecto.
- Contienen información de una fuente externa que ve nuestro producto sin prejuicios, el usuario potencial o real.
- Fomentan el trabajo en grupo en busca de soluciones.
- Permiten estimar el esfuerzo que va a requerir desarrollar una idea.

De lo que se trata es de invitar a la reflexión y fomentar la conversación en torno a resolver una problemática concreta del usuario o del proyecto, tanto de manera interna como incluyendo a personas ajenas a la empresa. (64)

A continuación, se muestra la historia de usuario del requisito funcional “Valorar un curso”. Para ver las restantes historias de usuario remitirse al Anexo 1: Historias de usuario.

Tabla 5. Historia de usuario del requisito funcional “Valorar un curso”.

Número: 2		Nombre del requisito: Valorar un curso	
Programador: Andy González Rosabal		Iteración Asignada: 1ra	
Prioridad: Alta		Tiempo Estimado: 4 días	
Riesgo en Desarrollo: N/A		Tiempo Real: 3 días	
Descripción: Permite al usuario realizar una valoración a un curso. El usuario puede valorar un curso al hacer click en alguna de las 5 estrellas que representan valores de 0-5 con un nivel de aceptación: malo para 1, pobre para 2, regular para 3, bueno para 4 y maravilloso para 5.			
Observaciones: El usuario para valorar un curso debe estar autenticado en el sistema.			
Prototipo de interfaz:			
			

2.4 Análisis y diseño

Análisis y diseño es una disciplina de la metodología de desarrollo de software AUP-UCI. Considera que los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de

estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Además, en esta disciplina se modela el sistema y su forma para que soporte todos los requisitos, incluyendo los requisitos no funcionales. (42)

Modelo del análisis

El objetivo del modelo del análisis es describir los dominios de información, función y comportamiento que se requieren para un sistema basado en computadoras. Utiliza una combinación de formatos en texto y diagramas para representar los requisitos de los datos, las funciones y el comportamiento de una manera que es relativamente fácil de entender y, aún más importante, conduce a una revisión para lograr la corrección, la integridad y la consistencia. (65)

2.4.1 Diagrama de clases del análisis

A continuación, se muestra el diagrama de clases del análisis del requisito funcional “Valorar un curso”. Para ver los diagramas de los restantes requisitos remitirse al Anexo 2: Diagramas de clases del análisis.

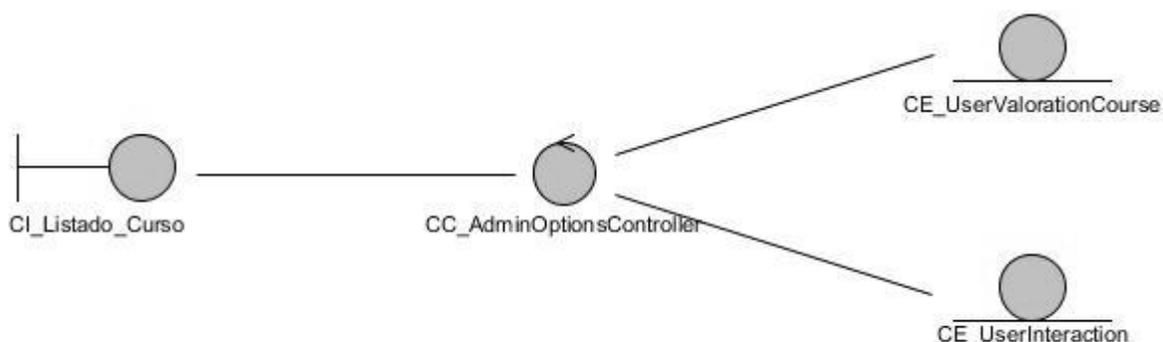


Figura 6. Diagrama de clases del análisis “Valorar un curso”

2.4.2 Diagrama de colaboración del análisis

El modelo del análisis utiliza un diagrama de colaboración para mostrar la forma en la que las clases de análisis colaboran una con la otra. (65)

A continuación, se muestra el diagrama de colaboración del análisis del requisito funcional “Valorar un curso”. Para el estudio de los restantes diagramas remitirse al Anexo 3: Diagramas de colaboración del análisis.

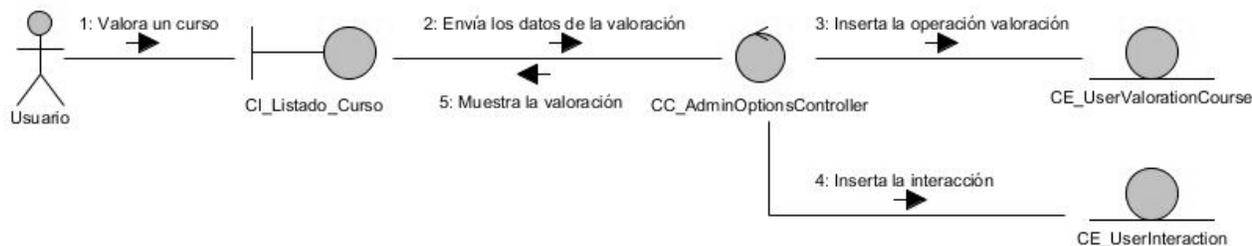


Figura 7. Diagrama de colaboración del análisis “Valorar un curso”

Diseño

El diseño del software implica identificar y describir las abstracciones fundamentales del sistema de software y sus relaciones. Es la toma de decisiones sobre la organización lógica del software. A menudo, se representa esta organización lógica como un modelo en un lenguaje definido de modelado como UML y, otras veces, simplemente se utiliza notaciones informales y esbozos para representar al diseño. (63)

2.4.3 Arquitectura de software

Una arquitectura de software define la estructura del sistema, la cual adquiere gran importancia debido a que las representaciones de arquitecturas de software permiten la comunicación entre todas las partes interesadas en el desarrollo de un sistema de cómputo. Dicha arquitectura constituye un modelo comprensible de cómo está estructurado el sistema y cómo trabajan juntos sus componentes. (66)

Arquitectura Cliente-Servidor

En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, produce una demanda de información a cualquiera de las computadoras que proporcionan información, conocidas como servidores estos últimos responden a la demanda del cliente que la produjo. Mediante esta arquitectura el usuario puede acceder a la información sin tener en cuenta su ubicación física y donde pueda estar alojada la misma. (65)

2.4.4 Patrón arquitectónico

Los patrones arquitectónicos, o patrones de arquitectura, son patrones de diseño de software que ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema, que consta de subsistemas, sus responsabilidades e interrelaciones (67). Para el desarrollo del sistema, se utiliza como patrón arquitectónico a seguir Modelo Vista Controlador, patrón en que está basado Symfony.

Patrón Modelo Vista Controlador

Dentro de las tecnologías a utilizar en el desarrollo del sistema se encuentra el framework Symfony el cual emplea el patrón arquitectónico Modelo-Vista-Controlador (MVC). Véase **Figura 8**.

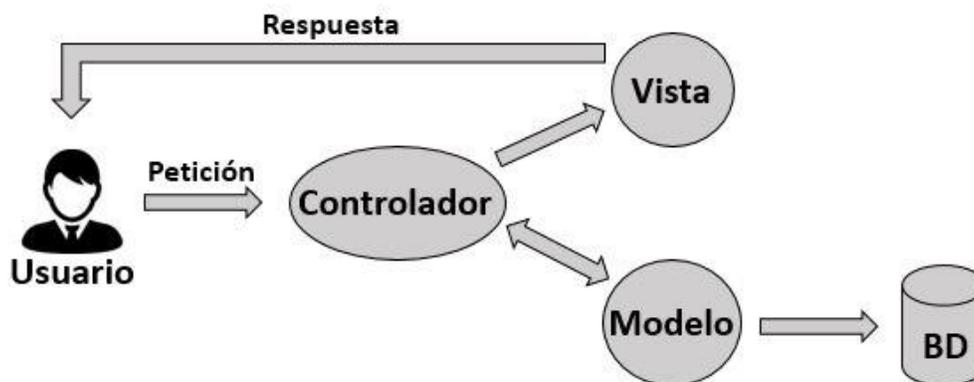


Figura 8. Patrón Modelo Vista Controlador. Elaboración propia

En resumen, el patrón de diseño MVC define una manera de organizar el código de acuerdo a su naturaleza. Este patrón separa el código en tres capas:

1. **Modelo:** Se refiere a la capa de acceso a datos. Esta capa contiene todo lo referido a los datos: cómo acceder a ellos, cómo validarlos, qué comportamiento tienen y las relaciones entre ellos.
2. **Vista:** Transforma el Modelo en una página web que permite al usuario interactuar con ella.
3. **Controlador:** Es la pieza de código que llama al Modelo para obtener algunos datos que le pasa a la Vista para la presentación al cliente.

2.4.5 Diagrama de clases del diseño

Un diagrama de clases del diseño muestra las definiciones de clases que posteriormente se implementarán en el sistema. Además, a diferencia del modelo conceptual, este diagrama no muestra gráficamente conceptos del mundo real; describe únicamente los componentes del software. (61)

A continuación, se muestra el diagrama de clases del diseño para el requisito funcional “Valorar un curso”. Para el estudio de los restantes diagramas remitirse al Anexo 4: Diagramas de clases del diseño.

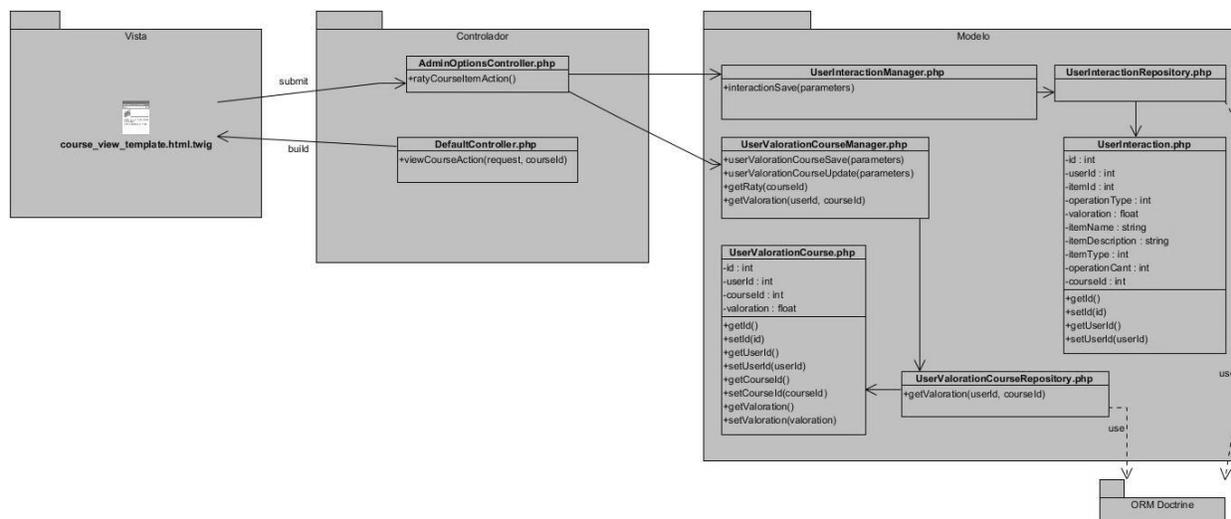


Figura 9. Diagrama de clases del diseño.

2.4.6 Diagrama de secuencia del diseño

Los diagramas de secuencia en el UML se usan principalmente para modelar las interacciones entre los actores y los objetos en un sistema, así como las interacciones entre los objetos en sí. El UML tiene una amplia sintaxis para diagramas de secuencia, lo cual permite muchos tipos diferentes de interacción a modelar. Muestra la sucesión de interacciones que ocurre durante un caso de uso particular o una instancia de caso de uso. (63)

A continuación, se muestra el diagrama de secuencia del requisito funcional “Valorar un curso”. Para el estudio de los restantes diagramas remitirse al Anexo 5: Diagramas de secuencia del diseño.

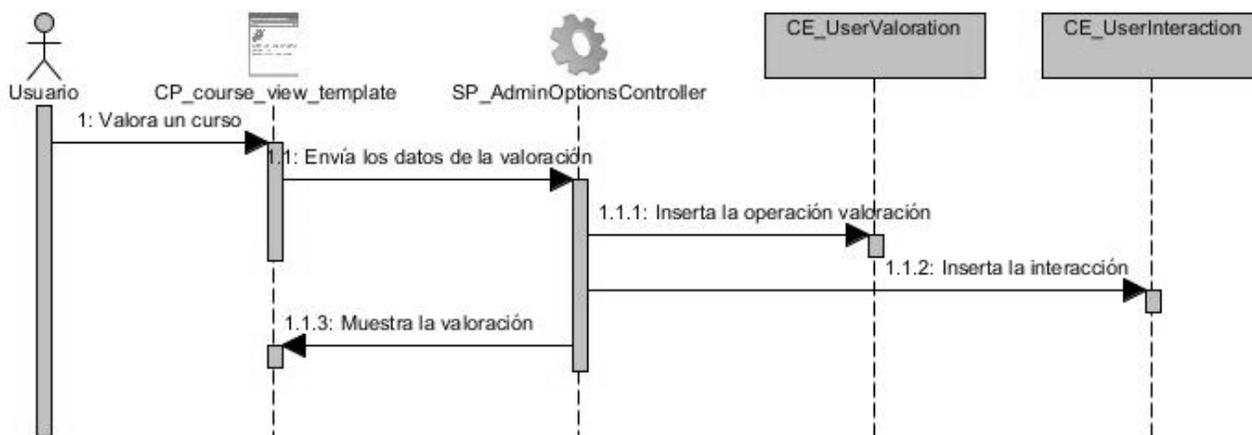


Figura 10. Diagrama de secuencia “Valorar un curso”

En el diagrama anterior, la vista `course_view_template` se asocia a la clase controladora `AdminOptionsController` del módulo creado, cuando en realidad pertenece al módulo `CourseBundle`, ya que se hace una reutilización de la misma; es decir, para lograr que el usuario valore los cursos y

almacenar en BD la información relacionada con la valoración, se hizo necesario reutilizar y añadir nuevas funcionalidades a la vista antes mencionada, funciones que van a ser ejecutadas en el controlador del módulo creado.

2.4.7 Patrones de diseño

Un patrón de diseño es una solución bien probada a un problema común que conjunta experiencia y buena práctica en una forma que pueda reutilizarse. Es una representación abstracta que puede ejemplificarse en varias formas. (63)

“Un patrón de diseño evita “reinventar la rueda” o, peor aún, inventar una “nueva rueda” que sea un poco menos redonda, demasiado pequeña para el uso que se pretende y muy angosta para el terreno en el que rodará.” (65)

Patrones GRASP

Los patrones generales de software para la asignación de responsabilidades (GRASP, por sus siglas en inglés de General Responsibility Assignment Software Patterns) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Son un método de enseñanza que ayuda a entender el diseño de POO, aplica al análisis y diseño de un modo sistemático, racional y explicable. (61)

La arquitectura de Symfony está diseñada con el objetivo de que se utilicen los cinco patrones de diseño que componen esta clasificación (Controlador, Creador, Alta cohesión, Bajo acoplamiento y Experto) situándolos en las capas de Modelo y Control que plantea el patrón arquitectónico MVC.

A continuación, se especifican los patrones que se evidencian en el módulo propuesto.

Experto: su funcionamiento consiste en asignar la responsabilidad al experto en la información, es decir, a la clase que contiene la información necesaria para cumplir la responsabilidad. De esta forma se logra que las clases tengan un mejor comportamiento y hacen que las mismas sean más cohesivas con el fin de darle soporte. Este patrón se evidencia en las clases `function.js` y `main.js`, los cuales utilizan el módulo `socket.io` para establecer las conexiones a cada uno de los servidores.

Creador: en la clase `Actions` se encuentran las acciones definidas para el sistema y se ejecutan en cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades, lo que evidencia que la clase `Actions` es “creador” de dichas entidades. Ejemplos de algunas funciones utilizadas en la clase `Actions` son: `doSelect ()`, `retrieveByPK ()`, `doSelectOne ()`.

Alta Cohesión: se evidencia cuando las responsabilidades de una clase están altamente relacionadas con las responsabilidades de otra clase. Este se evidencia entre la relación de las clases

AdminOptionsController y UserInteractionManager, ya que AdminOptionsController utiliza las funcionalidades que brinda la clase UserInteractionManager y a la vez esta última es la encargada de realizar las funcionalidades para trabajar en la base de datos, por lo que AdminOptionsController puede trabajar con la información de la base de datos sin una gran cantidad de trabajo. Además, este patrón se evidencia en la clase Actions, la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las properties.

Bajo Acoplamiento: este patrón mide el grado en que una clase está conectada a otra, tiene conocimiento de otra, o de alguna manera depende de otra. Es evaluativo, ya que un bajo acoplamiento permite que el diseño de clases sea más independiente, reduce el impacto de los cambios y aumenta la reutilización. Se evidencia en las clases app.js y server.js, debido a que en estos se puede observar un acoplamiento de control, que se produce cuando un módulo le envía elementos de control a otro módulo y determina la lógica de ejecución del mismo. Además, la clase Actions hereda únicamente de sfActions para alcanzar un bajo acoplamiento de clases.

Controlador: es un objeto responsable del manejo de los eventos del sistema, que no pertenece a la interfaz de usuario, el controlador recibe la solicitud del servicio desde la capa GUI, coordina su realización y delega en otros objetos. Este se evidencia en la clase AdminOptionsController, ya que es el encargado de controlar las peticiones del cliente. También en las clases sfFrontController, sfWebFrontController, sfContext y en los "actions" se evidencia el uso de este patrón, dado que ellas manejan eventos del sistema, tanto de lado del cliente como del servidor. (61)

Patrones GOF

Los patrones GOF (por sus siglas en inglés de The Gang of Four) describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos. (68)

Singleton: permite el manejo de objetos únicos y que sean accesibles a otros objetos. Además, permite el acceso controlado a una única instancia. Symfony usa por defecto este patrón al crear las instancias de los servicios, un ejemplo específico es el servicio fortes_uservalorationcourse.manager, que tiene como una de sus funciones devolver una instancia de la clase UserValorationCourse.

2.4.8 Diseño de la base de datos

Se le denomina diseño de la base de datos al boceto de las estructuras del sistema de datos y cómo se representarán en una base de datos. De nuevo, el trabajo aquí depende de si una base de datos se reutilizará o se creará una nueva. (63)

El Diseño Conceptual parte de los requisitos del cliente, lo que se pretende en esta fase es crear un esquema gráfico que permita identificar los elementos del problema y las relaciones que existen entre cada elemento, para el diseño se utilizan formas, por ejemplo: óvalos, rectángulos, líneas que describen las relaciones que existen entre cada elemento, un ejemplo de este diseño es el modelo entidad-relación. (69)

Un diagrama Entidad-Relación (ER), también conocido como modelo entidad relación, es un tipo de diagrama de flujo que ilustra cómo las "entidades", como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema. Los diagramas ER se usan a menudo para diseñar o depurar bases de datos relacionales en los campos de ingeniería de software, sistemas de información empresarial, educación e investigación. (70)

El modelo entidad relación de la Plataforma Educativa ZERA cuenta con un conjunto de paquetes y estos, a su vez, cuentan con una cierta cantidad de tablas. Con el desarrollo del módulo se agregan las entidades UserInteraction y UserValorationCourse. A continuación, se muestra el diagrama entidad relación con las tablas utilizadas en el desarrollo del módulo.

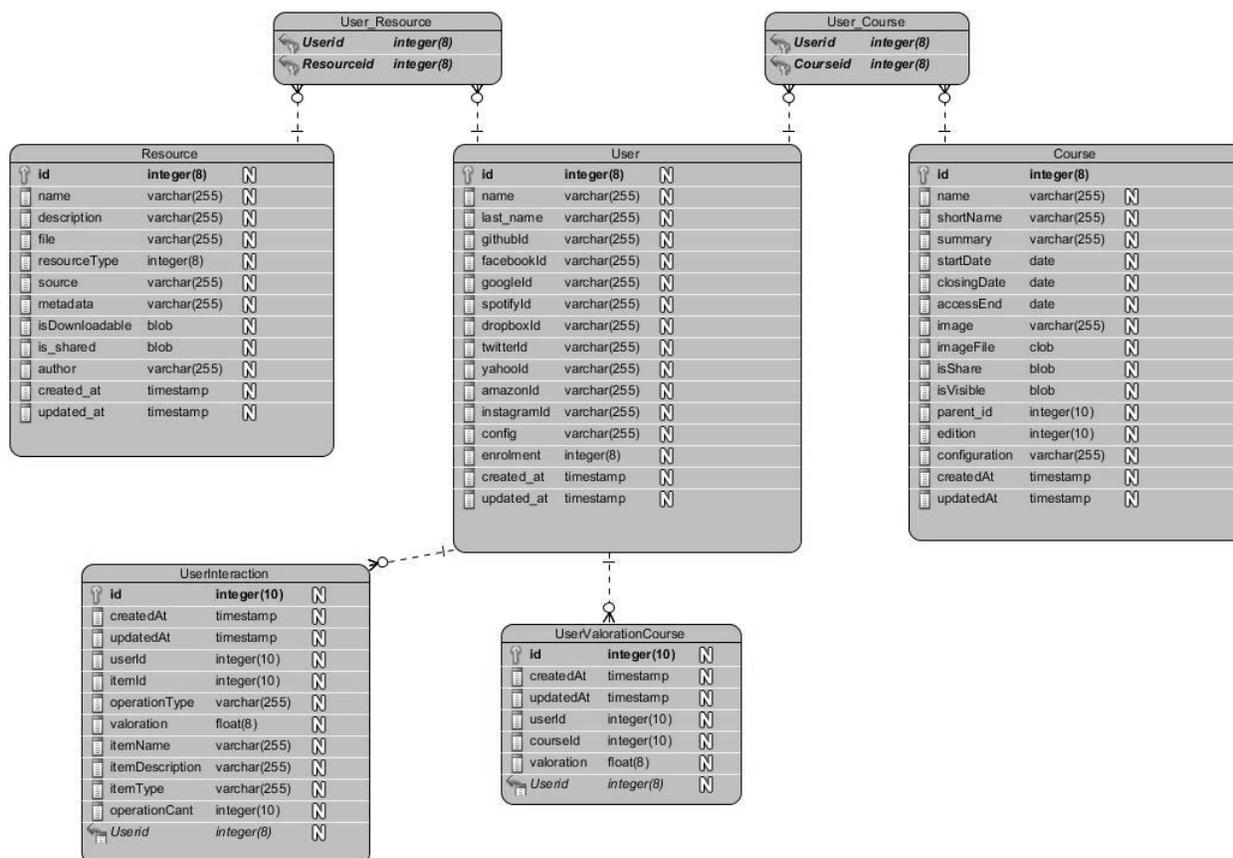


Figura 11. Modelo Entidad Relación.

Descripción de las tablas de la base de datos

En esta sección se presenta una breve descripción de cada uno de los atributos de la tabla “UserInteraction”. Para ver el resto de las descripciones, remitirse al Anexo 6: Descripciones de las tablas de la base de datos.

Tabla 6. Descripción de la entidad “UserInteraction” de la base de datos.

UserInteraction		
Descripción: En la presente tabla se muestran los atributos de la entidad UserInteraction.		
Atributo	Tipo	Descripción
id	integer(10)	Etiqueta única que identifica la interacción.
userId	integer(10)	Almacena el identificador del usuario que realizó la interacción.
itemId	integer(10)	Almacena el identificador del ítem interactuado por el usuario.
operationType	varchar(255)	Almacena el tipo de operación que hizo el usuario.
valuation	float(8)	Almacena la valoración de un usuario a un curso.
itemName	varchar(255)	Almacena el nombre del ítem interactuado por el usuario.
itemDescription	varchar(255)	Almacena la descripción del ítem interactuado por el usuario.
itemType	varchar(255)	Almacena el tipo de ítem interactuado por el usuario.
operationCant	integer(10)	Almacena la cantidad de veces que un usuario hizo determinada operación.
createdAt	timestamp	Almacena la fecha de la interacción.
updatedAt	timestamp	Actualiza la fecha de la interacción.

Conclusiones parciales

En el presente capítulo quedó establecida la propuesta de solución, se elaboró el modelo de dominio, donde se explicaron los conceptos significativos entorno al problema planteado. Se especificaron los requisitos funcionales y no funcionales de software, que sirvieron de guía para desarrollar las distintas funcionalidades y así satisfacer las necesidades del cliente. También, se realizaron los artefactos definidos en la fase de análisis y diseño de la metodología AUP–UCI en su escenario 4, constituyendo una guía fundamental para la construcción de la propuesta de solución. Además, se agregaron las tablas UserInteraction y UserValorationCourse a la base de datos de la Plataforma Educativa ZERA.

Capítulo 3. Implementación y prueba

La elaboración de la propuesta de solución presentada en el capítulo anterior, ayuda al programador a entender mejor las funcionalidades que busca el cliente y, por tanto, ser capaz de llevarlas a un lenguaje entendible por las computadoras. En el presente capítulo se abordan los aspectos más importantes referentes a la implementación y a las pruebas del sistema, con el objetivo de asegurar la calidad de la aplicación y que cumpla con todas las peticiones del cliente. Como principales elementos en este capítulo se definen los diagramas de componentes y de despliegue. Además, se enuncian los estándares de codificación que se utilizaron para generar el código fuente y se especifican los casos de pruebas aplicados a la solución desarrollada para validar su correcto funcionamiento.

3.1 Implementación

La implementación de un software incluye cómo desarrollar y organizar los componentes basándose en la fase de análisis y diseño, específicamente en los artefactos generados en la misma. Describe cómo los elementos del modelo de diseño se implementan en términos de componentes, en otras palabras, toma el resultado del modelo de diseño para generar el código final del sistema. Además, en la implementación se definen los estándares de codificación a emplear, se realizan las implementaciones a las historias de usuarios, entre otras actividades.

3.1.1 Estándares de código

Un estándar de codificación comprende todos los aspectos de la generación de código. Un código fuente completo debe de reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Los estándares de codificación permiten una mejor integración entre las líneas de producción y establecen pautas que conllevan a lograr un código más legible y reutilizable, de tal forma que se pueda aumentar su capacidad de almacenamiento a lo largo del tiempo. (71)

En la implementación de la aplicación se utilizaron diferentes estándares que se describen a continuación:

Definición de las clases

Las declaraciones de clases tienen su llave de apertura una línea más abajo de la declaración y el nombre de la clase comienza con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará la notación PascalCasing (cada palabra comienza con letra mayúscula). En el caso particular de las clases controladoras después del nombre llevan la palabra “Controller”. Se utiliza *namespaces* para todas las clases. Ver ejemplo en la figura 12.

```

namespace FORTES\RecommenderBundle\Controller;

class AdminOptionsController extends Controller
{
    ...
}
    
```

Figura 12. Definición de la clase AdminOptionsController. Elaboración propia.

Definición de las funciones

Los nombres de las funciones están precedidos por la palabra reservada que define el encapsulamiento del método (public, protected o private), seguido de la palabra function y después el nombre de la función. El nombre a emplear para las funciones se escribe con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará la notación **CamelCasing** (es parecido al *PascalCasing* con la excepción que la letra inicial del identificador debe estar en minúscula), y con solo leerlo se sabe el propósito de la misma. Ver ejemplo en la figura 13.

```

public function getNeighbors($userId, $cant)
{
    ...
    return $neighbors;
}
    
```

Figura 13. Definición de la función getNeighbors. Elaboración propia.

Definición de las variables

El nombre de las variables se escribe con la primera letra en minúscula, si es un nombre compuesto se empleará la notación **CamelCasing** descrita anteriormente. Siempre las variables llevarán delante del nombre el símbolo de peso (“\$”) como establece el lenguaje de programación PHP, en el caso de la declaración de variables en el lenguaje JavaScript, se utiliza la palabra reservada (“var”) antes del nombre. Ver ejemplo en la figura 14.

```

public function getCoursesRecommendation($cantCourses, $cantNeighbors)
{
    $cant = 0;
    $coursesToRecommender = array();
}
    
```

Figura 14. Definición de variables. Elaboración propia.

Definición de los comentarios

Los comentarios deben ser precisos y cortos de forma tal que se entienda el propósito de lo desarrollado. Se ubican encima de la función a la cual hace referencia dicho comentario. Los comentarios que abarcan una sola línea se definen con doble barra (“//”). Los bloques de comentarios se definen con barra-asterisco

para iniciar el bloque (“/*”) y con asterisco-barra para finalizarlo (“*/”). Ver figura 15. Además, se utilizan las anotaciones siguientes:

@param: provee el nombre, el tipo y la descripción de los parámetros de una función.

@var: define qué tipo de dato se representa mediante la propiedad.

@return: documenta el valor que retorna una función.

@Route: ruta de donde accede a los datos necesarios para la implementación de la función.

@Method: método por el cual captura los datos la función.

```

/**
 * @Route("/video/play", name="play_video_item", options={"expose"=true})
 */
public function playVideoItemAction()
{
    ...
}

/**
 * @var integer $id
 */
private $id;

/**
 * @param \DateTime $createdAt Fecha de la operacion hecha por el usuario.
 *
 * @return UserInteraction
 */
public function setCreatedAt(\DateTime $createdAt)
{
    //$this->createdAt = $createdAt;
    /*
    return $this;
    */
}
    
```

Figura 15. Definición de comentarios. Elaboración propia.

Asignación de variables y llamadas a funciones

Las asignaciones de variables se realizarán mediante el signo de igualdad (“=”) o también mediante el símbolo de flecha (“->”), se coloca un espacio alrededor del operador. Las llamadas a funciones se realizarán sin espacios entre el nombre de la función y los paréntesis, los atributos pasados por parámetro se separarán por una coma y un espacio en caso de existir más de un atributo. Ver ejemplo en la figura 16.

```

public function getContentsRecommendation()
{
    $cont = 0;
    $neighbors = $this->getNeighbors($realUserId, $scantNeighbors);
    ...
}
    
```

Figura 16. Asignación de variables y llamadas a funciones. Elaboración propia.

Estructuras de control

Las estructuras de control utilizadas en la implementación son: *if*, *if else*, *for* y *while*. Se adicionará un espacio alrededor de cada operador binario (“==”, “!=”, “&&”, “||”, “<=”, “>=”, ...), con excepción del operador de concatenación punto (“.”). Ver ejemplo en la figura 17.

```

public function getRaty($courseId)
{
    ...

    if ($raty != null){
        $valoracion = $raty->getValoracion();
        return $valoracion;
    }else{
        return null;
    }
}
    
```

Figura 17. Estructura de control if else. Elaboración propia.

3.1.2 Diagrama de componentes

El diagrama de componentes muestra las partes internas de un software y las relaciones que existen entre ellas, indica cómo un sistema de software está constituido por un conjunto de componentes desplegados que interactúan a través de interfaces bien definidas y que tienen sus detalles internos ocultos. (72)

A continuación, se muestra el diagrama de componentes:

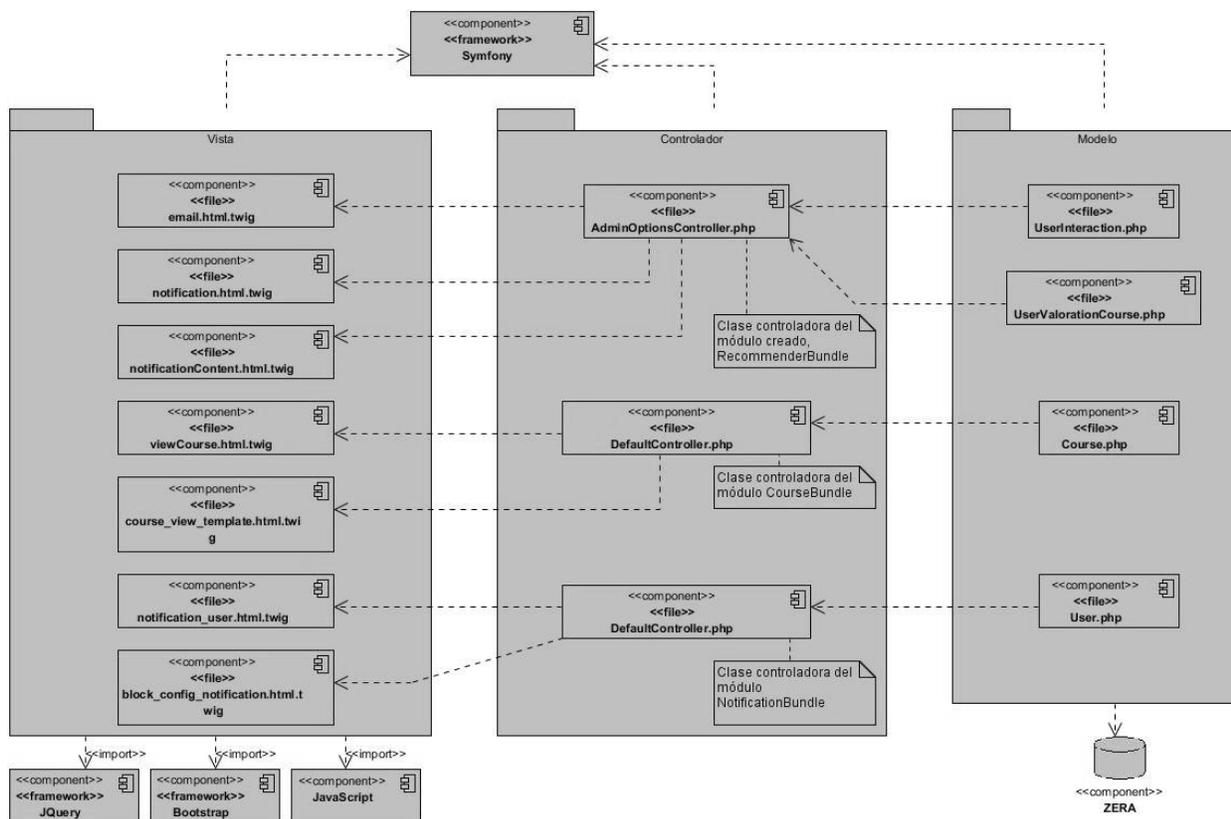


Figura 18. Diagrama de componentes.

En el diagrama anterior se muestran los componentes DefaultController.php como parte de la solución propuesta, es válido aclarar que dichas clases controladoras no fueron implementadas por el autor de la investigación, sino que ya formaban parte de la aplicación y se encuentran en los módulos CourseBundle y NotificationBundle. La solución propuesta no solo está encapsulada en el módulo creado, por ejemplo, para poder almacenar en BD la interacción de los usuarios con los cursos y contenidos de la plataforma, se hizo necesario implementar otras funcionalidades en vistas ya existentes como viewCourse y course_view_template que pertenecen al módulo CourseBundle; por este motivo, se incluyen en el diagrama de componentes vistas que no están ubicadas en el módulo creado, pero si forman parte de la solución.

3.1.3 Diagrama de despliegue

Un diagrama de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución y los componentes que viven en ellos. Los diagramas de despliegue son un tipo de diagrama estructurado utilizado en el modelado de los aspectos físicos de un sistema orientado a objetos. A menudo se utilizan para modelar la vista de implementación estática de un sistema. (73)

A continuación, se muestra el diagrama de despliegue del sistema.

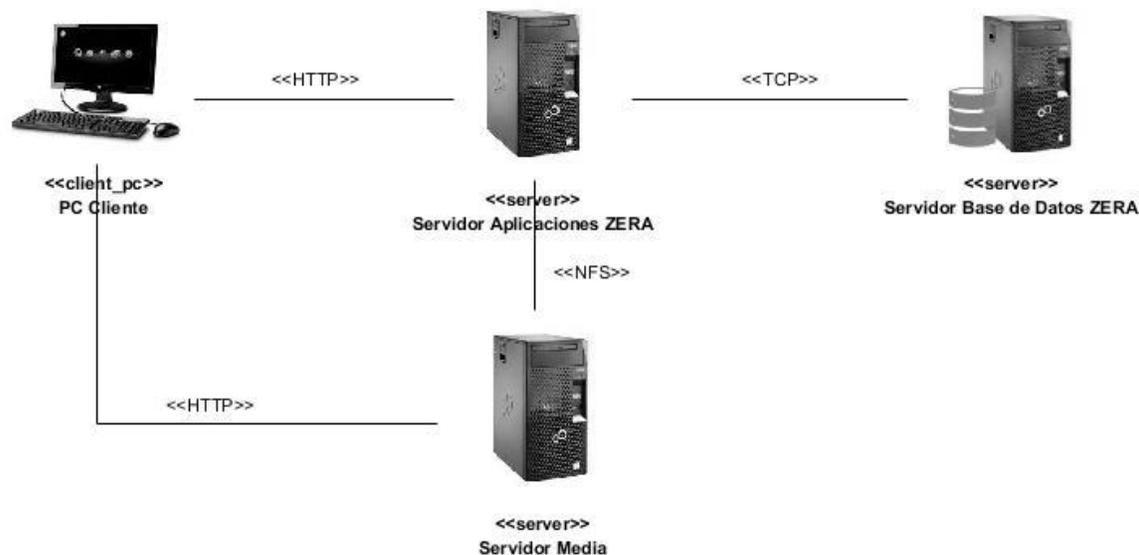


Figura 19. Diagrama de despliegue. Elaboración propia.

El diagrama de despliegue representa la distribución física del sistema a través de los siguientes nodos:

- ✓ **PC Cliente:** espacio de trabajo que el usuario utilizará para acceder a la aplicación web e interactuar con el sistema. Deberá tener instalado un navegador web.
- ✓ **Servidor Aplicaciones ZERA y Servidor Media:** son utilizados para la publicación de la aplicación. Son la herramienta principal para ejecutar la lógica de negocio en el lado del servidor y el código de las páginas servidor. El servidor web utilizado es Apache.
- ✓ **Servidor Base de Datos ZERA:** hace referencia al gestor de bases de datos donde se encuentran los datos necesarios para el trabajo con el sistema. El servidor de base de datos elegido es PostgreSQL.

Un protocolo de comunicación es un conjunto de reglas establecidas entre dos dispositivos para permitir la comunicación entre ambos.

- ✓ **Protocolo HTTP:** es el Protocolo de Transferencia de Hipertexto (HTTP, por sus siglas en inglés de HyperText Transfer Protocol), utilizado entre los navegadores de los clientes y el servidor web. El elemento HTTP de la arquitectura representa un tipo de comunicación no orientado a la conexión entre clientes y servidor.
- ✓ **Protocolo TCP:** es el Protocolo de Control de Transmisión (TCP, por sus siglas en inglés de Transmission Control Protocol), utilizado entre el servidor web y el servidor de base de datos. Garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron.
- ✓ **Protocolo NFS:** el Sistema de Archivos de Red (NFS, por sus siglas en inglés de Network File System), es un protocolo que permite el acceso remoto a un sistema de archivos a través de la

red. El protocolo NFS es uno de varios estándares de sistema de archivos distribuidos para almacenamiento atado a la red. (74)

3.2 Pruebas de software

Las pruebas de software intentan demostrar que un programa hace lo que se espera que haga, así como descubrir defectos en el programa antes de usarlo. Es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones de los requisitos del sistema. El proceso de pruebas tiene dos objetivos distintivos: (63)

1. Demostrar al desarrollador y al cliente que el software satisface sus requisitos. Para el software personalizado, esto significa que debería haber al menos una prueba para cada requerimiento o característica que se incorporará a la entrega del producto.
2. Descubrir situaciones donde el comportamiento del software sea incorrecto, no deseable o no cumple su especificación. La prueba de defectos está relacionada con la eliminación de todos los tipos de comportamientos del sistema no deseables, tales como caídas del sistema, interacciones no permitidas con otros sistemas, cálculos incorrectos y corrección de datos.

A continuación, se describen los tipos de pruebas aplicadas al módulo desarrollado:

Pruebas funcionales: las pruebas funcionales se definen como un tipo de prueba, verifican que cada función de la aplicación de software se ejecuta de acuerdo con la especificación del requisito correspondiente. A las pruebas funcionales también se les denomina pruebas de comportamiento o de caja negra, debido a que el probador enfoca su atención en las respuestas del sistema y no le preocupa el código fuente de la aplicación. Cada una de las funcionalidades del sistema se prueban al proporcionar una entrada adecuada, verificar la salida y comparar los resultados reales con los resultados esperados. Implican, entre otros elementos, la comprobación de la interfaz de usuario, las API y la base de datos. Las pruebas se pueden realizar manualmente o mediante automatización. (75)

Pruebas de carga y estrés: la prueba de carga es un tipo de prueba que determina el rendimiento de un sistema en condiciones de carga reales, ayuda a determinar cómo se comporta la aplicación cuando varios usuarios acceden a ella simultáneamente. Identifica la capacidad máxima de operación de una aplicación, determina si la infraestructura actual es suficiente para ejecutar la aplicación y si la aplicación es sostenible con respecto a la carga máxima de usuarios. Por otra parte, las pruebas de estrés se utilizan normalmente para “romper” la aplicación, se aumenta cada vez más el número de usuarios y se ejecuta una prueba de carga hasta que se rompe, se lleva a cabo para estar seguros que el sistema no se bloquea en situaciones difíciles. (76)

Pruebas de regresión: las pruebas de regresión se refieren al proceso de prueba de un programa informático modificado o actualizado para asegurarse de que las funciones de software más antiguas, que fueron desarrolladas y probadas anteriormente, funcionan exactamente como lo hacían antes. La prueba de regresión no es más que una selección total o parcial de casos de prueba ya ejecutados que se vuelven a ejecutar para garantizar que las funcionalidades existentes funcionen bien. Se realiza para asegurarse que los nuevos cambios de código no deben tener efectos secundarios en las funcionalidades existentes. Asegura que el código anterior aún funcione una vez que se realicen los cambios en el nuevo código. (77)

Pruebas de aceptación: las pruebas de aceptación se definen como un tipo de prueba realizada por el cliente para certificar el sistema con respecto a los requisitos acordados. Se realizan en la fase final de las pruebas antes de mover la aplicación de software al mercado o entorno de producción. Es un tipo de prueba de caja negra en la que participarán dos o más usuarios finales que validen la conformidad con los requisitos. (78)

3.2.1 Pruebas de caja blanca

Las pruebas de caja blanca, en ocasiones llamada pruebas de vidrio, es un método de diseño que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de pruebas. Al emplear lo métodos de prueba de caja blanca, el ingeniero de software podrá derivar casos de prueba que: (65)

- Garanticen que todas las rutas independientes dentro del módulo se revisaron al menos una vez.
- Se ejerciten los lados verdaderos y falsos de todas las decisiones lógicas.
- Se ejecuten todos los bucles en sus fronteras y dentro de sus límites operacionales.
- Se revisen estructuras de datos internas para garantizar su validez.

3.2.2 Pruebas de caja negra

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requisitos funcionales del software. Las pruebas de caja negra permiten al ingeniero de software derivar conjuntos de condiciones de entradas que revisarán por completo todos los requisitos funcionales de un software. No son una alternativa para las pruebas de caja blanca, son un enfoque complementario donde es probable descubrir errores diferentes de los que se descubrirán con los métodos de caja blanca. Intentan encontrar errores como: (65)

- Funciones incorrectas o faltantes.
- De interfaz.
- En las estructuras de datos o en el acceso a bases de datos externas.

- De comportamiento o rendimiento.
- De inicialización o terminación.

Con el objetivo de aplicar las pruebas de caja negra o funcional, es necesario apoyarse en el diseño de casos de prueba propuesto por la metodología de desarrollo de software seleccionada. Un caso de prueba es una vía para comprobar el buen funcionamiento del sistema, en ellos se incluyen las entradas, resultados y condiciones que se han de probar, constituyendo la guía principal para el probador.

3.2.3 Diseño de casos de prueba

Al momento de diseñar un caso de prueba, en primer lugar, se selecciona una característica del sistema o componente que se desee probar. A continuación, se eligen un conjunto de entradas que ejecutan dicha característica, documenta las salidas esperadas o rangos de salida y, donde sea posible, se diseña una prueba automatizada que valida que las salidas reales y las esperadas son las mismas. (63)

A continuación, se muestra el diseño de casos de prueba para el requisito funcional “Configurar notificaciones de tipo recomendación”. En el Anexo 7: Diseño de casos de prueba, se encuentran los restantes casos de prueba identificados.

Tabla 7. Diseño de casos de prueba para el RF “Configurar notificaciones de tipo recomendación”.

Escenario	Descripción	Datos de entrada	Respuesta del sistema	Flujo central
EC 1.1 Opción configurar notificaciones de tipo recomendación.	Selecciona la opción Configurar Notificaciones.	N/A	Brinda la posibilidad al usuario de configurar las notificaciones de tipo recomendación. Puede seleccionar recibir recomendación en pantalla, mediante correo electrónico, ambas o ninguna.	Perfil de usuario/ Notificaciones/ Configurar notificaciones
EC 1.2 Opción recibir recomendación en pantalla.	Selecciona la opción Plataforma y hace click en el botón Actualizar.	N/A	Actualiza el archivo json del perfil de usuario con las nuevas preferencias, muestra el mensaje: "Se ha realizado la acción satisfactoriamente". El usuario solo desea recibir notificaciones de tipo recomendación en pantalla.	Perfil de usuario/ Notificaciones/ Configurar notificaciones/ Actualizar
EC 1.3 Opción recibir recomendación vía correo electrónico.	Selecciona la opción Correo electrónico y hace click en el botón Actualizar.	N/A	Actualiza el archivo json del perfil de usuario con las nuevas preferencias, muestra el mensaje: "Se ha realizado la acción satisfactoriamente". El usuario solo desea recibir notificaciones de tipo recomendación vía correo electrónico.	Perfil de usuario/ Notificaciones/ Configurar notificaciones/ Actualizar
EC 1.4 Marcar ambas opciones.	Selecciona la opción Correo electrónico, la opción Plataforma y hace click en el botón Actualizar.	N/A	Actualiza el archivo json del perfil de usuario con las nuevas preferencias, muestra el mensaje: "Se ha realizado la acción satisfactoriamente". El usuario podrá recibir notificaciones de tipo recomendación por ambas vías.	Perfil de usuario/ Notificaciones/ Configurar notificaciones/ Actualizar
EC 1.5 Dejar de marcar ambas opciones.	No selecciona ninguna opción y hace click en el botón Actualizar.	N/A	Muestra un mensaje de error con la siguiente información: "Seleccione Estrategia para Almacenar". <u>Regresa al EC 1.1</u>	Perfil de usuario/ Notificaciones/ Configurar notificaciones/ Actualizar

3.2.4 Resultados obtenidos

Resultados de las pruebas funcionales

Al módulo recomendador de cursos y contenidos en la Plataforma Educativa ZERA, le fue aplicado pruebas funcionales para detectar las no conformidades presentes en la propuesta de solución. A través del método de caja negra, se realizaron 3 iteraciones, las cuales arrojaron los siguientes resultados:

Primera Iteración: se encontraron 26 no conformidades clasificadas en 20 de complejidad alta, 2 de complejidad media y 4 de complejidad baja.

Segunda Iteración: resultaron 5 no conformidades clasificadas en 3 de complejidad alta, 1 de complejidad media y 1 de complejidad baja.

Tercera Iteración: no se detectaron no conformidades.

En total, se detectaron 31 no conformidades después de haber concluido la ejecución de las pruebas, de las cuales fueron resueltas las 31. A continuación, se muestra un gráfico de barra con los resultados obtenidos.

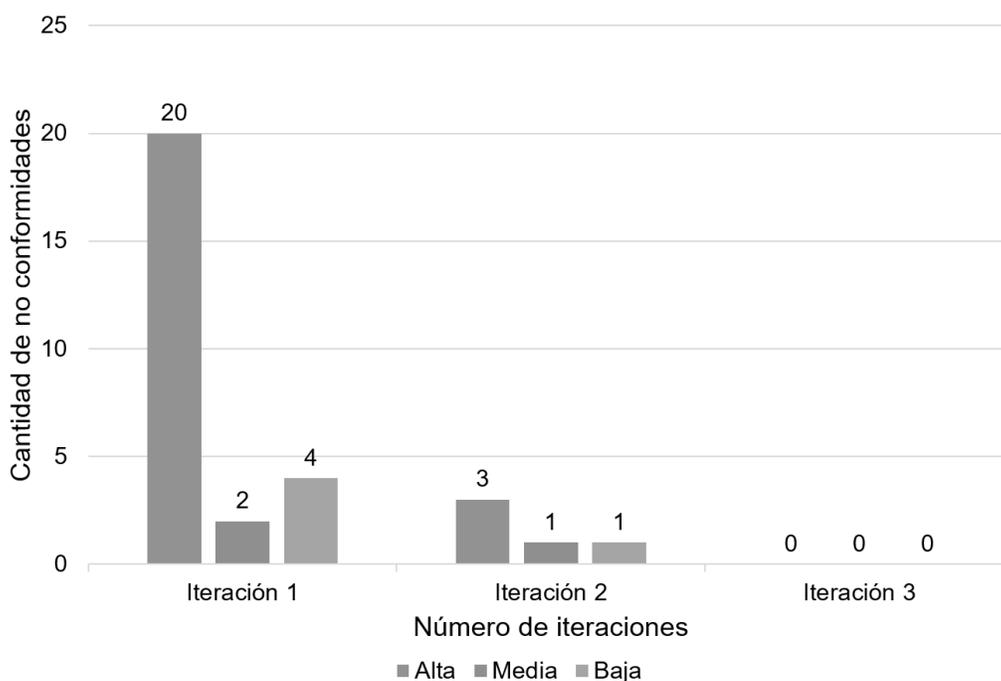


Gráfico 1. Cantidad de no conformidades identificadas por las pruebas funcionales.

Resultados de las pruebas de carga y estrés

Para la realización de las pruebas de carga y estrés, se utilizó la herramienta Apache JMeter, la cual se describe a continuación:

Apache JMeter es una herramienta de prueba de carga de código abierto, basada en Java, que se puede utilizar para analizar el comportamiento funcional de un sistema y medir el rendimiento de un sistema bajo una prueba de carga. Se puede usar para simular cargas variables o pesadas en servidores, redes u objetos singulares o múltiples para probar la fortaleza de un sistema. JMeter simula grupos de usuarios que envían solicitudes a un servidor o red, y luego devuelve las estadísticas a un usuario a través de diagramas visuales. La interfaz gráfica de Apache JMeter es similar a un navegador; sin embargo, no puede representar páginas HTML o el JavaScript que se encuentra en las páginas como lo haría un navegador. Además, admite el uso de complementos. (79)

Características del ordenador utilizado para el desarrollo de las pruebas de carga y estrés:

- Microprocesador Intel Celeron.
- Sistema operativo Linux Mint 18.2 Cinnamon 64-bit.
- Memoria RAM 4GB.

En correspondencia con las características del ordenador empleado, los resultados que arrojaron las pruebas de carga se consideran buenos, ya que los tiempos de respuesta del servidor se encuentran en un rango entre 2 y 5 segundos. A continuación, se presenta un gráfico con el rendimiento obtenido para un total de 10, 30 y 50 usuarios conectados al mismo tiempo.

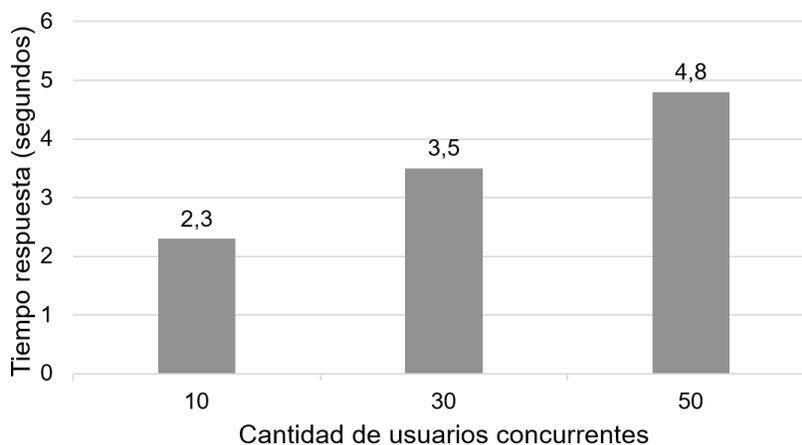


Gráfico 2. Resultados obtenidos para la prueba de carga.

Al igual que las pruebas de carga, los resultados obtenidos de las pruebas de estrés se consideran buenos, debido a que, después de elevar el número de usuarios conectados al mismo tiempo a 100, 140 y 200, el módulo desarrollado permaneció estable con sus funcionalidades sin incurrir en fallos. A continuación, se presentan los resultados obtenidos una vez aplicadas las pruebas de estrés al módulo desarrollado:

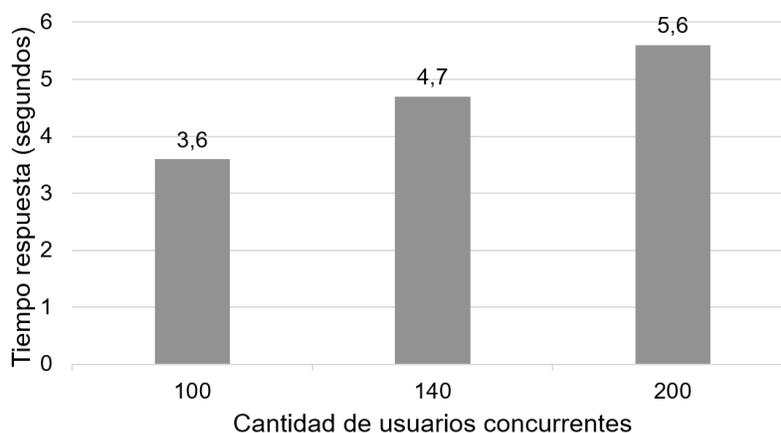


Gráfico 3. Resultados obtenidos para la prueba de estrés.

Una vez ejecutadas las pruebas, como resultado se obtuvo un conjunto de no conformidades, algunas de ellas se muestran en la siguiente tabla:

Tabla 8. No conformidades detectadas al módulo implementado.

No. NC	Requisito Funcional	Descripción	Complejidad	Estado
1	RF 1	No se guarda en BD cuando el usuario accede a un curso.	Alta	Resuelta
2	RF 1	Cuando el usuario visualiza un documento, se guarda en BD como si hubiera visto una imagen.	Alta	Resuelta
3	RF 2	Cuando el usuario valora un curso y se recarga la página, se pierde la valoración.	Alta	Resuelta
4	RF 2	Solo se puede valorar un curso en números enteros, no se puede valorar con decimales, 3.5 por ejemplo.	Media	Resuelta
5	RF 2	Cuando el usuario valora un curso que ya había valorado anteriormente, no se actualiza la valoración en BD.	Alta	Resuelta
6	RF 5,6,7,8	Faltan términos por definir en los archivos de traducción.	Baja	Resuelta
7	RF 6	El campo Recomendación no tiene acento ortográfico.	Baja	Resuelta
8	RF 7	La notificación en pantalla se retira muy rápido (a los pocos segundos).	Media	Resuelta
9	RF 7	La notificación en pantalla no se muestra en idioma inglés.	Baja	Resuelta

10	RF 7	Los links de los cursos en la notificación en pantalla están rotos, no llevan al usuario a ninguna parte.	Alta	Resuelta
11	RF 8	No se envía el correo electrónico a los estudiantes.	Alta	Resuelta
12	RF 8	No se muestran las imágenes de los cursos en los correos.	Alta	Resuelta
13	RF 8	No se envía el correo, el sistema muestra el mensaje: "El SwiftMailer no está bien configurado".	Alta	Resuelta
14	RF 6	Cuando el usuario señala que no desea recibir notificaciones de tipo recomendación en pantalla, no se actualizan las preferencias en BD.	Alta	Resuelta

Conclusiones parciales

En el presente capítulo se definieron los estándares de codificación empleados en la implementación del Módulo para recomendar al usuario cursos y contenidos en la Plataforma Educativa ZERA. Se establecieron estándares para la definición de los métodos, variables y comentarios, para la implementación de las clases, entre otros. La utilización de los estándares de codificación, permitió adoptar una estructura homogénea que facilita la comunicación y asegura la calidad, menor cantidad de errores y fácil mantenimiento. También se crearon los diagramas de componentes y de despliegue, lo que permitió observar la estructura general del sistema y de la topología de hardware donde se ejecuta el mismo. Además, se le realizaron diferentes pruebas al módulo implementado para garantizar su calidad, entre ellas se encuentran las pruebas funcionales, ejecutadas a través del método de caja negra, detectándose un total de 31 no conformidades, de las cuales fueron resueltas las 31. Se realizaron las pruebas de carga y estrés, las cuales arrojaron tiempos de respuestas entre 2 y 6 segundos.

Conclusiones generales

En el presente trabajo de diploma se realizó la investigación, el análisis, el diseño, la implementación y las pruebas a un módulo que permite recomendar cursos y contenidos en la Plataforma Educativa ZERA. De esta forma se le da cumplimiento al objetivo general de la investigación presentada y se pudo arribar a las siguientes conclusiones:

- Los estudios teóricos y metodológicos de la investigación, demostraron la utilidad e importancia de los sistemas de recomendación para la web actual, en campos como el comercio y el aprendizaje electrónico donde se destacan las plataformas educativas.
- La utilización de la metodología AUP-UCI, facilitó la implementación del módulo recomendador mediante los artefactos generados, entre los que se encuentran las historias de usuario, los diagramas de clases del diseño y el modelo entidad-relación.
- La realización del análisis y diseño según la metodología AUP-UCI en su escenario 4, permitió obtener un total de 7 requisitos funcionales y 5 no funcionales.
- El uso de un conjunto de patrones y buenas prácticas de programación, permitió diseñar e implementar diferentes clases que dan cumplimiento a los requisitos funcionales y no funcionales de la propuesta de solución.
- Se obtuvo un módulo funcional que responde a los requisitos planteados y a las necesidades del cliente, que facilitará a los usuarios de la Plataforma Educativa ZERA, el acceso a cursos y contenidos de su interés de acuerdo a su interacción y a las de otros usuarios con el sistema.
- La ejecución de las pruebas funcionales al módulo implementado, permitió detectar 31 no conformidades, las cuales fueron resueltas en 3 iteraciones; mientras que las pruebas de carga y estrés, se consideran satisfactorias en correspondencia con las características del ordenador en que fueron ejecutadas, ya que arrojaron tiempos de respuestas entre 2 y 6 segundos.

Recomendaciones

Una vez terminada la implementación del módulo y analizados los resultados obtenidos en la presente investigación, se mantiene la idea de mejorar la solución final, por lo que se recomienda al equipo de desarrollo de la Plataforma Educativa ZERA:

- Incorporar al módulo desarrollado, la posibilidad de recomendar al usuario, otros recursos educativos como son los foros, seminarios y videoconferencias.
- Añadir al módulo implementado, la funcionalidad de interactuar con el Sistema de Gestión Universitaria (SGU) de la UCI, con el propósito de conocer las asignaturas suspensas de un determinado usuario para tenerlas en cuenta a la hora de hacerle la recomendación.
- Implementar en el módulo desarrollado, un SR basado en contenidos, con el fin de incorporarle las bondades de un sistema híbrido.

Referencias Bibliográficas

1. **Pupo Fernández, Dallany** . Sistemas de Recomendación. [En línea] 6 de octubre de 2015. [Citado el: 5 de febrero de 2019.] <https://revista.jovenclub.cu/sistemas-de-recomendacion/>. 1995-9419.
2. **Martínez Hernández, Aliosky**. Los Sistemas Recomendadores de recursos de información. [En línea] 13 de enero de 2015. [Citado el: 2 de febrero de 2019.] <https://www.monografias.com/trabajos104/sistemas-recomendadores-recursos-informacion/sistemas-recomendadores-recursos-informacion.shtml>.
3. **Guzmán, Augusto**. *Sistema de Recomendación y Planificación Turística de la ciudad de Valencia vía web*. s.l. : Universidad de Valencia, 2009.
4. **Aula1**. Plataformas educativas ¿Qué son y para qué sirven? [En línea] 16 de enero de 2017. [Citado el: 20 de marzo de 2019.] <https://www.aula1.com/plataformas-educativas/>.
5. **Font, Miguel**. Sistemas de recomendación para webs de información sobre la salud. [En línea] Universidad Politécnica de Cataluña, 2009. [Citado el: 22 de noviembre de 2018.] <http://upcommons.upc.edu/pfc/handle/2099.1/7193>.
6. **Bertate, Alicia, Machado, Rodrigo y Molina, Valeri**. PGMúsica Sistema de Recomendación de música. [En línea] Uruguay: Universidad de la Republica, 2016. [Citado el: 15 de noviembre de 2018.] <http://www.fing.edu.uy/inco/grupos/pln/prygrado/InformePGMusica.pdf>.
7. **Anderson, Chris**. *The Long Tail*. 2004.
8. **Martínez, Carlos Javier**. Implantación y evaluación de un recomendador social de canciones. [En línea] Universidad Politécnica de Valencia, 2014. [Citado el: 11 de octubre de 2018.] <https://riunet.upv.es/bitstream/handle/10251/53954/MART%C3%8DNEZ%20-%20Implantaci%C3%B3n%20y%20evaluaci%C3%B3n%20de%20un%20recomendador%20social%20de%20canciones.pdf?sequence=3>.
9. **Moya, Ricardo**. ¿Que son los Sistemas de Recomendación? [En línea] Jarroba, 2014. [Citado el: 11 de octubre de 2018.] <http://jarroba.com/que-son-los-sistemas-de-recomendacion/>.
10. **Herlocker, J.L, y otros**. Evaluating Collaborative Filtering Recommender Systems. [En línea] ACM Transactions on Information Systems Vol. 22, 2004. [Citado el: 16 de noviembre de 2018.] <http://groupLens.org/site-content/uploads/evaluating-TOIS-20041.pdf>.
11. **Jannach, Dietmar**. *Recommender Systems: An Introduction*. s.l. : Cambridge University Press, 2012. 978-051-176-3113.
12. **Castellano Torres, Emilio Jose**. *Evaluación del uso de algoritmos colaborativos para orientar académicamente al alumnado en Bachillerato*. s.l. : Universidad de Jaen, 2007.
13. **Adomavicius, G y Tuzhilin, A**. *Recommendation Technologies: Survey of current methods and possible extensions*. s.l. : University of Minnesota : MISRC, 2003.
14. **De Gemmis, Marco y Laquinta, Leo**. *Preference Learning in a Recommender System*. Slovenia : s.n., 2009.
15. **Rodríguez, Marko y Allen, David**. *A recommender system to support the scholarly communication process. The Computing Research Repository*. 2009.

16. **Vozalis, Emmanouil y Margaritis, Konstantinos.** *Analysis of Recommender Systems' Algorithms. The 6th Hellenic European Conference on Computer Mathematics & its Applications.* Atenas, Grecia : s.n., 2003.
17. **Swearingen, Kirsten y Sinha, Rashmi.** *Beyond Algorithms: An HCI Perspective on Recommender Systems. ACM SIGIR 2001 Workshop on Recommender Systems.* 2001.
18. **Peña, Fernando Andrés y Riffo, Ricardo Elías.** *Revisión, selección e implementación de un algoritmo de recomendación de material bibliográfico utilizando tecnología j2ee.* Chile : Universidad del Bío-Bío, 2008.
19. **Peinado, Andrea Silva, y otros.** Diseño e implementación de un software recomendador y adaptativo de educación básica secundaria en las instituciones educativas del municipio de Lorica-Cordoba. [En línea] Buenas Tareas, 2011. [Citado el: 14 de octubre de 2018.] <http://www.buenastareas.com/ensayos/Dise%C3%B1o-e-Implementaci%C3%B3n-De-Un-Software/2714517.html>.
20. **Pazzani, Michael y Billsus, Daniel.** *Content-Based Recommendation System. The Adaptive Web: Methods and Strategies of Web Personalization.* Springer Berlin / Heidelberg : s.n., 2007.
21. **SINBAD, Grupo de Investigación.** Personalización y los Sistemas de Recomendación. [En línea] Universidad de Jaen, 2012. [Citado el: 23 de octubre de 2018.] <http://sinbad2.ujaen.es/cod/archivosPublicos/presentaciones/sistemasDeRecomendacion.pdf>.
22. **Linden, Greg, Smith, Brent y York, Jeremy.** *Recommendations: Item-to-Item Collaborative Filtering.* s.l. : IEEE Internet Computing, 2013. págs. 76-80.
23. **Ruiz Almudena, Iniesta.** Estrategias de recomendación aplicadas a repositorios de recursos educativos. [En línea] E-Prints Complutense, 2009. [Citado el: 20 de octubre de 2018.] <http://eprints.ucm.es/9908>.
24. **Burke, Robin.** *Hybrid recommender systems: Survey and experiments.* s.l. : California State University, 2002. Vol. 12.
25. **Sobecki, Janusz.** *Implementations of Web-based Recommender Systems Using Hybrid Methods.* s.l. : International Journal of Computer Science & Applications, 2006. págs. 52-64.
26. **Moya, Ricardo.** Sistemas de Recomendación basados en Filtrado Colaborativo (K-Vecinos). [En línea] 24 de septiembre de 2013. [Citado el: 3 de mayo de 2019.] <https://jarroba.com/sistemas-de-recomendacion-basados-en-filtrado-colaborativo-k-vecinos/>.
27. **Srivastava, Tavish .** Introduction to k-Nearest Neighbors: Simplified (with implementation in Python). [En línea] 26 de marzo de 2018. [Citado el: 9 de febrero de 2019.] <https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>.
28. **Uberoi, Anannya .** K-Nearest Neighbours. [En línea] s.f. [Citado el: 20 de marzo de 2019.] <https://www.geeksforgeeks.org/k-nearest-neighbours/>.
29. **Gorreta, L.** Análisis de Cluster, un ejemplo sencillo. [En línea] 18 de marzo de 2015. [Citado el: 3 de mayo de 2019.] <https://dlegorreta.wordpress.com/tag/distancia-euclidiana/>.
30. **Metcalf, Leigh.** Metrics, similarity, and sets. [En línea] 2016. [Citado el: 3 de mayo de 2019.] <https://www.sciencedirect.com/topics/computer-science/euclidean-distance>.

31. **Ben Schafer, J y Konstan, J.** *Recommender Systems in E-Commerce ACM Conference on Electronic Commerce*. 2009. págs. 158-166.
32. **Tan, Huiyi, Guo, Junfei y Li, Yong.** *E-Learning Recommendation System. IEEE International Conference on Computer Science and Software Engineering*. 2008. págs. 431-434.
33. **García Salcines, Enrique y Romero Morales, Cristóbal.** *Sistema recomendador colaborativo usando minería de datos distribuida para la mejora continua de cursos e-learning*. 2008. págs. 19-30.
34. **Tang, Y y McCalla, G.** *Smart recommendations for an evolving e-learning system. Workshop on Technologies for Electronic Documents for Supporting Learning*. 2003.
35. **Bertate, Leticia, Machado, Rodrigo y Molina, Valeria.** PGMúsica Sistema de Recomendación de música. [En línea] 2006. [Citado el: 18 de octubre de 2018.] <http://www.fing.edu.uy/inco/grupos/pln/prygrado/InformePGMusica.pdf>.
36. **Herrera-Viedma, Enrique, Porcel, Carlos y Hidalgo, Lorenzo.** Sistemas de recomendaciones: herramientas para el filtrado de información en Internet. [En línea] España. Universidad de Granada, 2004. [Citado el: 15 de noviembre de 2018.] <https://www.upf.edu/hipertextnet/numero-2/recomendacion.html#3.5>.
37. **Moreno, Pablo.** 10 plataformas de cursos online gratuitos. [En línea] 3 de enero de 2017. [Citado el: 18 de febrero de 2019.] https://fueracodigos.com/10-plataformas-cursos-online-gratuitos/#2_Coursera.
38. **CENGAGE.** 37 Plataformas virtuales educativas gratuitas. [En línea] 14 de marzo de 2016. [Citado el: 19 de febrero de 2019.] <http://latinoamerica.cengage.com/37-plataformas-virtuales-educativas-gratuitas/>.
39. **Carrillo, Isaías, Pérez, Rodrigo y Rodríguez, Aureliano David.** *Metodología de Desarrollo de Software*. 2008.
40. **ProcesosDeSoftware.** Metodologías para desarrollo de software. [En línea] 2015. [Citado el: 15 de diciembre de 2018.] <http://procesosdesoftware.wikispaces.com/METODOLOGIAS+PARA+DESARROLLO+DE+SOFTWARE...>
41. **Figuroa, R y Solís, C.** *Metodologías Tradicionales Vs Metodologías Ágiles*. s.f.
42. **Rodríguez Sánchez, Tamara.** *Metodología de desarrollo para la Actividad productiva de la UCI*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2015.
43. **Tendenzias.** Qué es un entorno de desarrollo integrado, IDE. [En línea] 2015. [Citado el: 10 de octubre de 2018.] <http://programaciondesarrollo.es/que-es-un-entorno-de-desarrollo-integrado-ide>.
44. **Grupo Características.** 5 características de Netbeans. [En línea] 31 de agosto de 2018. [Citado el: abril de 26 de 2019.] <https://www.12caracteristicas.com/netbeans/>.
45. **Fabien, Potencier.** *Symfony, la Guía Definitiva*. 2007.
46. **Álvarez, Miguel A.** *Manual de jQuery*. 2016.
47. **Fontela, Álvaro.** ¿Qué es Bootstrap? [En línea] 2016. [Citado el: 23 de noviembre de 2018.] <https://raiolanetworks.es/blog/que-es-bootstrap/>.
48. **Robledano, Ángel M.** Tutorial Bootstrap 3: Introducción e instalación. [En línea] 2015. [Citado el: 24 de noviembre de 2018.] <https://openwebinars.net/tutorial-bootstrap-3-introduccion-e-instalacion/>.

49. **Larman, Craig.** *Introducción al análisis y diseño orientado a objetos.* 2003.
50. **NGINX.** [En línea] s.f. [Citado el: 9 de diciembre de 2018.] <https://www.nginx.com>.
51. **Python Software, Foundation.** Python Software. [En línea] 2012. [Citado el: noviembre de 22 de 2018.] <http://www.python.org/>.
52. **García de Jalón, Javier.** *Aprenda Java como si estuviera en primero.* San Sebastián : s.n., 2000.
53. **Stewart, Celeste.** The advantage of PHP. Designer's Playground. [En línea] 2011. [Citado el: 22 de octubre de 2018.] <http://www.designersplayground.com/articles/118/1/The-Advantages-of-PHP/Page1.html>.
54. **Mah, Paul.** Disadvantages of Web Development Using PHP. ITBusinessEdge. [En línea] 2009. [Citado el: 22 de noviembre de 2018.] <http://www.itbusinessedge.com/cm/blogs/mah/disadvantages-of-web-development-using-php/?cs=33397>.
55. **Eguíluz Pérez, Javier.** *Introducción a XHTML.* 2013.
56. **De Luca, Damian.** CSS3 HTML5 . [En línea] 2010. [Citado el: 5 de diciembre de 2018.] <http://html5.dwebapps.com/que-es-css3/>.
57. **Pérez, Damián.** Maestros del Web. [En línea] 2017. [Citado el: 21 de noviembre de 2018.] <http://www.maestrosdelweb.com/que-es-javascript/>.
58. **UML.** UML. [En línea] 2011. [Citado el: 15 de octubre de 2018.] <http://www.uml.org>.
59. **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* Prentice Hall : s.n., 1999. ISBN 0-13-748880-7.
60. **Moquillaza, Santiago Domingo, Vega, Hugo y Guerra, Luis.** *Programación en N capas.* s.l. : RISI, 2010. Vol. 7.
61. **Larman, Craig.** UML y Patrones. 2da. s.l. : Prentice Hall, 2003.
62. **Grupo Carricay.** ¿Cómo escribir un buen documento de especificación de requisitos de software? [En línea] 25 de junio de 2018. [Citado el: 6 de marzo de 2019.] <https://medium.com/grupo-carricay/c%C3%B3mo-escribir-un-buen-documento-de-especificaci%C3%B3n-de-requisitos-de-software-fd8bb3b5a39a>.
63. **Sommerville, Ian.** *INGENIERÍA DE SOFTWARE.* 9na. México : Pearson, 2011. 978-607-32-0603-7.
64. **Solving Ad Hoc.** Qué son las historias de usuario y su función en Agilidad. [En línea] 18 de diciembre de 2017 . [Citado el: 5 de marzo de 2019.] <https://solvingadhoc.com/las-historias-usuario-funcion-agilidad/>.
65. **Pressman, Roger S.** *Ingeniería del software. Un enfoque práctico.* 7ma ed. University of Connecticut : s.n., 2010. 978-607-15-0314-15.
66. **Cervantes, Humberto.** Arquitectura de Software. [En línea] 2018. [Citado el: 30 de noviembre de 2018.] <https://sg.com.mx/revista/27/arquitectura-software>.
67. **Jacobson, Ivar.** *El Proceso Unificado de Desarrollo de Software.* 2009.
68. **Guerrero, Carlos, Suárez, Johana y Gutiérrez, Luz.** Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. [En línea] 24 de noviembre de

2013. [Citado el: 11 de marzo de 2019.] https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-07642013000300012.
69. **Largo, Elivar**. Lenguaje SQL y Base de Datos. [En línea] 12 de marzo de 2017. [Citado el: 11 de marzo de 2019.] <https://www.ecodeup.com/disenio-de-base-de-datos-y-lenguaje-sql/>.
70. **Grupo Lucidchart**. Qué es un diagrama entidad-relación. [En línea] 11 de mayo de 2017. [Citado el: 11 de marzo de 2019.] <https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>.
71. **Microsoft Group**. Revisiones de código y estándares de codificación. [En línea] 12 de mayo de 2018. [Citado el: 22 de abril de 2019.] <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx...>
72. **Osis, Janis y Donins, Uldis**. Learn more about Component Diagram. *ScienceDirect*. [En línea] 1 de diciembre de 2018. [Citado el: 22 de abril de 2019.] <https://www.sciencedirect.com/topics/computer-science/component-diagram>.
73. **Visual Paradigm Group**. What is Deployment Diagram? *Visual Paradigm*. [En línea] 29 de septiembre de 2017. [Citado el: 22 de abril de 2019.] <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/>.
74. **Rouse, Margaret**. Sistema de archivos de red, NFS. [En línea] 1 de septiembre de 2017. [Citado el: 20 de febrero de 2019.] <https://searchdatacenter.techtarget.com/es/definicion/Sistema-de-archivos-de-red-NFS>.
75. **Guru Group**. What is Functional Testing? Types & Examples (Complete Tutorial) . [En línea] 2 de abril de 2019. [Citado el: 23 de abril de 2019.] <https://www.guru99.com/functional-testing.html>.
76. —. Load Testing Tutorial: What is? How to? (with Examples) . [En línea] 15 de marzo de 2019. [Citado el: 23 de abril de 2019.] <https://www.guru99.com/load-testing-tutorial.html>.
77. **Stringfellow, Angela**. What Is Regression Testing? Best Practices, Tutorials, and More. [En línea] 14 de abril de 2017. [Citado el: 23 de abril de 2019.] <https://dzone.com/articles/what-is-regression-testing-best-practices-tutorial>.
78. **Guru Group**. What is User Acceptance Testing (UAT)? with Examples . [En línea] 2 de abril de 2019. [Citado el: 23 de abril de 2019.] <https://www.guru99.com/user-acceptance-testing.html>.
79. **Rouse, Margaret**. Apache JMeter. Definition. [En línea] 11 de marzo de 2019. [Citado el: 26 de abril de 2019.] <https://searchsoftwarequality.techtarget.com/definicion/Apache-JMeter>.

Anexo 1: Historias de usuarios

Historia de usuario del RF "Interactuar con los cursos y contenidos"

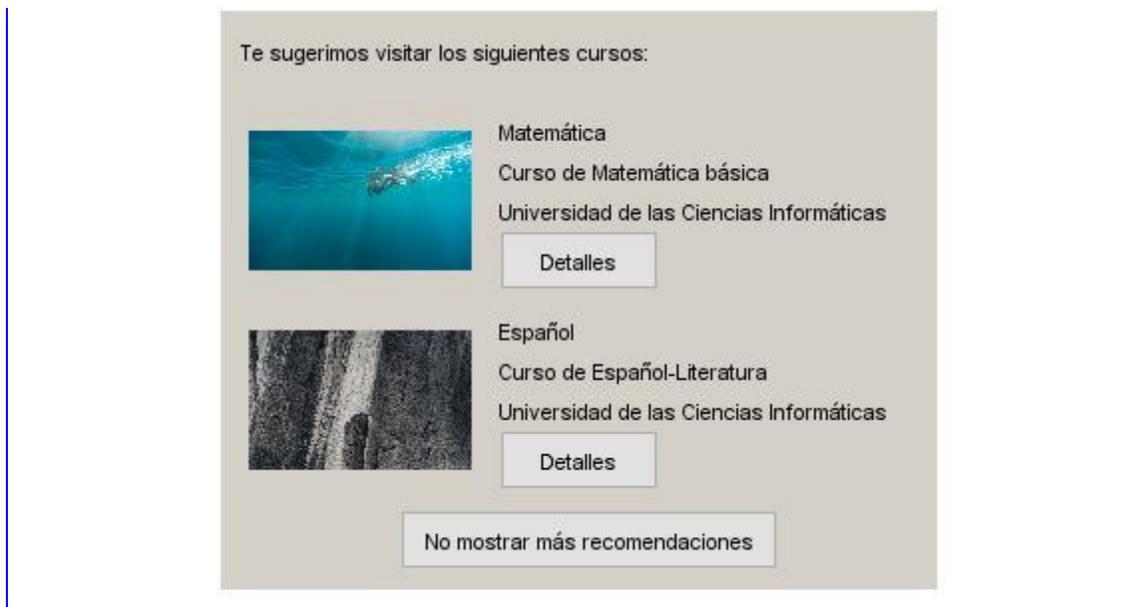
Número: 1		Nombre del requisito: Interactuar con los cursos y contenidos	
Programador: Andy González Rosabal		Iteración Asignada: 1ra	
Prioridad: Alta		Tiempo Estimado: 20 días	
Riesgo en Desarrollo: N/A		Tiempo Real: 18 días	
<p>Descripción:</p> <p>El sistema debe almacenar en base de datos la información sobre la interacción de los usuarios con los cursos y contenidos de la Plataforma Educativa ZERA. Es decir, cuando el usuario acceda a un curso, visualice una imagen, cuando reproduzca un audio o un video o cuando descargue un documento, se guarde el usuario que hizo la operación, el tipo de operación (visualizar, acceder, reproducir, descargar), el tipo de contenido (imagen, video, audio, documento, curso), el identificador del contenido, el curso al que está asociado dicho contenido, la cantidad de veces que realizó esa operación, entre otros elementos.</p>			
<p>Observaciones: El sistema debe actualizar la información sobre la interacción de los usuarios con los cursos y contenidos de la plataforma. Es decir, si un contenido cambia su nombre o descripción, se deberá actualizar en base de datos esa información, además, cada vez que un usuario repita una interacción ya guardada en base de datos, se actualizará la cantidad de veces que realizó esa interacción sumando 1 a la cantidad anterior.</p> <p>El usuario para interactuar con los cursos y contenidos debe estar autenticado en el sistema.</p>			
<p>Prototipo de interfaz. Ejemplo visualizar foto:</p> <div style="border: 1px solid #ccc; padding: 10px; width: fit-content; margin: 0 auto;">  </div>			

Historia de usuario del RF "Valorar un curso"

Número: 2		Nombre del requisito: Valorar un curso	
Programador: Andy González Rosabal		Iteración Asignada: 1ra	
Prioridad: Alta		Tiempo Estimado: 4 días	
Riesgo en Desarrollo: N/A		Tiempo Real: 3 días	
<p>Descripción:</p> <p>Permite al usuario realizar una valoración a un curso. El usuario puede valorar un curso haciendo click en alguna de las 5 estrellas que representan valores de 0-5 con un nivel de aceptación: malo para 1, pobre para 2, regular para 3, bueno para 4 y maravilloso para 5.</p>			
<p>Observaciones: El usuario para valorar un curso debe estar autenticado en el sistema.</p>			
<p>Prototipo de interfaz:</p> <div style="text-align: center; background-color: #d3d3d3; padding: 10px; border: 1px solid #ccc;"> <p>Matemática</p> <p>6 de Marzo de 2019</p> <div style="border: 1px solid #ccc; padding: 5px; display: inline-block; margin: 5px 0;">Desmatricular</div> <p>Universidad de las Ciencias Informáticas</p>  </div>			

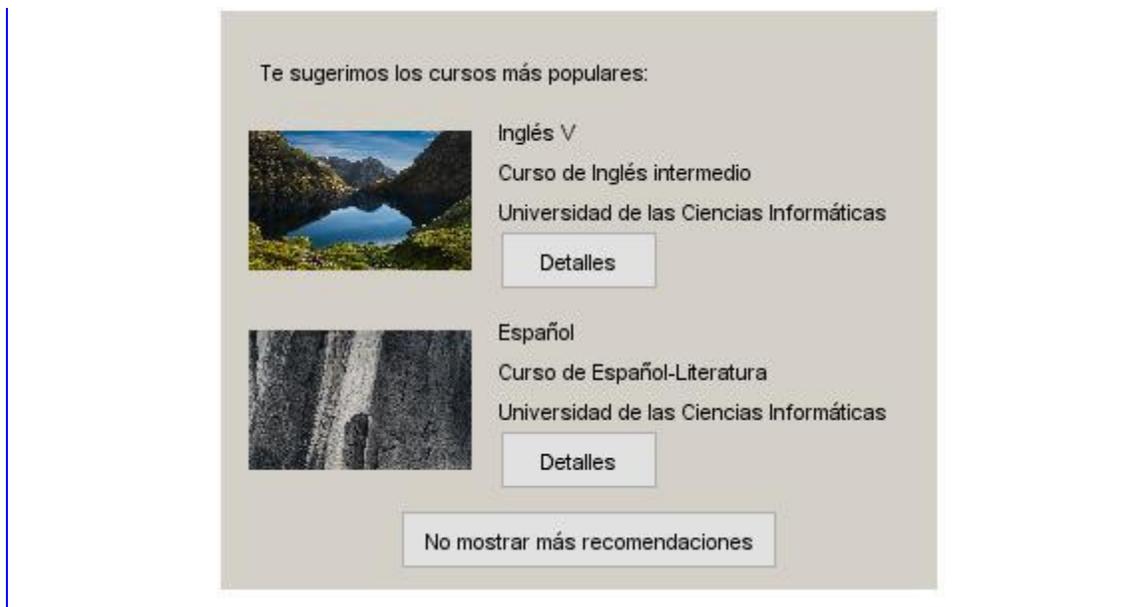
Historia de usuario del RF "Mostrar recomendación de cursos y contenidos obtenidos mediante el algoritmo de la vecindad próxima"

Número: 3	Nombre del requisito: Mostrar recomendación de cursos y contenidos obtenidos mediante el algoritmo de la vecindad próxima
Programador: Andy González Rosabal	Iteración Asignada: 1ra
Prioridad: Alta	Tiempo Estimado: 17 días
Riesgo en Desarrollo: N/A	Tiempo Real: 17 días
<p>Descripción:</p> <p>El sistema debe mostrar al usuario en pantalla, la recomendación de cursos y contenidos obtenidos mediante el algoritmo de la vecindad próxima. Se aplica el algoritmo, recibe como entrada el conjunto de datos recopilados con anterioridad sobre la interacción de los usuarios con los cursos y contenidos. Para un usuario determinado, se obtienen los K vecinos más cercanos. Una vez se cuente con los vecinos cercanos, para el caso de los cursos, se le recomienda al usuario aquellos que han sido mejores valorados por los vecinos cercanos y que él no se haya inscrito aún para podérselos recomendar; para el caso de los contenidos, se obtienen los más vistos (más accedidos) por los vecinos cercanos y se le recomiendan. La notificación en pantalla se muestra cada cierto tiempo. En el caso de la recomendación de cursos, el usuario puede hacer click en el botón Detalles para acceder a los detalles del curso y poder inscribirse. Si no desea recibir más sugerencias de cursos y contenidos, puede hacer click en el botón No mostrar más recomendaciones.</p>	
<p>Observaciones: El usuario debe estar autenticado en el sistema. El usuario debe permitir recibir notificaciones de tipo recomendación en pantalla.</p>	
<p>Prototipo de interfaz. Ejemplo sugerencia de cursos:</p>	



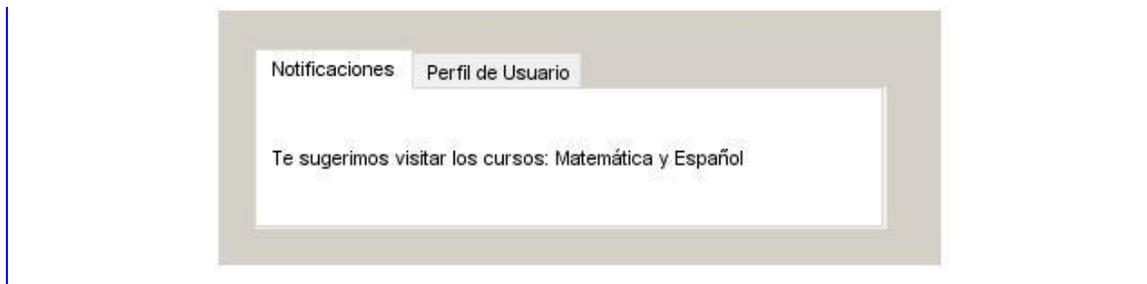
Historia de usuario del RF "Mostrar recomendación con los cursos mejores valorados"

Número: 4	Nombre del requisito: Mostrar recomendación con los cursos mejores valorados
Programador: Andy González Rosabal	Iteración Asignada: 1ra
Prioridad: Media	Tiempo Estimado: 9 días
Riesgo en Desarrollo: N/A	Tiempo Real: 8 días
Descripción: El sistema debe mostrar al usuario en pantalla, la recomendación con los cursos mejores valorados de la plataforma. Por cada curso, se calcula el promedio de las valoraciones hechas por los usuarios y se obtienen los cursos de mejores promedios (los mejores valorados) del sistema. La notificación en pantalla se muestra cada cierto tiempo. El usuario puede hacer click en el botón Detalles para acceder a los detalles del curso y poder inscribirse. Si no desea recibir este tipo de notificaciones, puede hacer click en el botón No mostrar más recomendaciones.	
Observaciones: El usuario debe estar autenticado en el sistema. Deben existir valoraciones a cursos de la plataforma. El usuario debe permitir recibir notificaciones de tipo recomendación en pantalla.	
Prototipo de interfaz:	



Historia de usuario del RF "Visualizar recomendación de cursos en el apartado Notificaciones de la plataforma"

Número: 5	Nombre del requisito: Visualizar recomendación de cursos en el apartado Notificaciones de la plataforma
Programador: Andy González Rosabal	Iteración Asignada: 1ra
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en Desarrollo: N/A	Tiempo Real: 4 días
Descripción: Una vez realizada la recomendación de cursos, el usuario puede recordar lo recomendado en la sección Notificaciones de la plataforma. Esta notificación se realiza tanto para los cursos obtenidos a partir del algoritmo KNN como para los cursos mejores valorados de la plataforma.	
Observaciones: El usuario debe estar autenticado en el sistema y haber recibido la recomendación en pantalla con anterioridad. El recordatorio en la sección Notificaciones es solo para los cursos, los contenidos solo se recomiendan en pantalla.	
Prototipo de interfaz:	



Historia de usuario del RF "Configurar notificaciones de tipo recomendación"

Número: 6	Nombre del requisito: Configurar notificaciones de tipo recomendación
Programador: Andy González Rosabal	Iteración Asignada: 1ra
Prioridad: Alta	Tiempo Estimado: 2 días
Riesgo en Desarrollo: N/A	Tiempo Real: 2 días
<p>Descripción:</p> <p>Permite al usuario elegir si desea o no recibir notificaciones de tipo recomendación vía correo electrónico, en pantalla, ambas o ninguna. El usuario puede seleccionar la notificación que desea recibir haciendo click en las opciones que se le muestran y luego, dar click en Actualizar para guardar las nuevas preferencias.</p>	
<p>Observaciones: El usuario para configurar las notificaciones de tipo recomendación debe estar autenticado en el sistema. Debe dirigirse a la pestaña Configurar notificaciones del perfil de usuario.</p>	
<p>Prototipo de interfaz:</p> 	

Historia de usuario del RF "Notificar recomendación de cursos vía correo electrónico"

Número: 7	Nombre del requisito: Notificar recomendación de cursos vía correo electrónico
Programador: Andy González Rosabal	Iteración Asignada: 1ra
Prioridad: Alta	Tiempo Estimado: 4 días
Riesgo en Desarrollo: N/A	Tiempo Real: 4 días
<p>Descripción:</p> <p>El sistema debe notificar al usuario vía correo electrónico la recomendación de cursos generada. El usuario desde el correo obtiene información de los cursos recomendados. La notificación vía correo electrónico se realiza cada cierto tiempo. El usuario puede hacer click en el botón Detalles para acceder a los detalles del curso y poder inscribirse. Si no desea recibir más correos con sugerencias, puede hacer click en el botón No mostrar más recomendaciones. Esta notificación se realiza tanto para los cursos obtenidos a partir del algoritmo KNN como para los cursos mejores valorados de la plataforma.</p>	
<p>Observaciones: El usuario debe permitir recibir notificaciones de tipo recomendación vía correo electrónico y estar autenticado. La notificación vía correo electrónico es para recomendar cursos, los contenidos solo se recomiendan en pantalla.</p>	
<p>Prototipo de interfaz:</p> <div style="border: 1px solid #ccc; padding: 10px; background-color: #f0f0f0;"> <p>De: ZERA 2.0</p> <p>Para: user@uci.cu</p> <p>CC:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <div style="display: flex; align-items: flex-start;">  <div> <p>Matemática</p> <p>Curso de Matemática Básica</p> <p>Universidad de las Ciencias Informáticas</p> <p style="text-align: right;">Detalles</p> </div> </div> <div style="display: flex; align-items: flex-start; margin-top: 10px;">  <div> <p>Álgebra Lineal</p> <p>Curso de Álgebra</p> <p>Universidad de las Ciencias Informáticas</p> <p style="text-align: right;">Detalles</p> </div> </div> <p style="text-align: center; margin-top: 10px;">No mostrar más recomendaciones</p> </div> </div>	

Anexo 2: Diagramas de clases del análisis

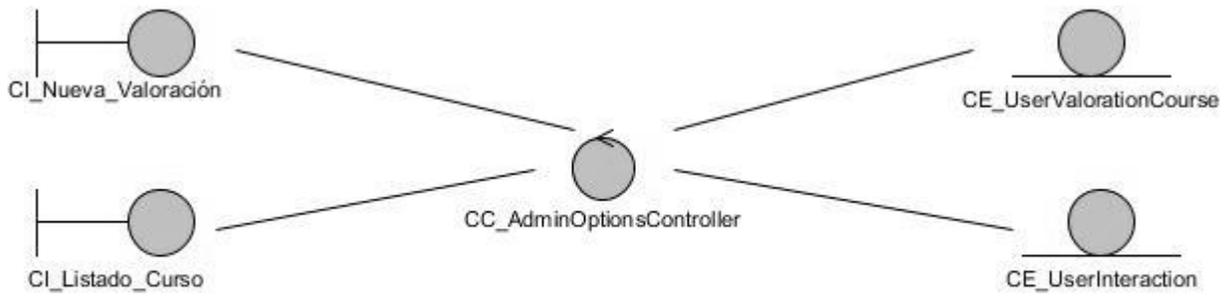


Diagrama de clases del análisis RF1

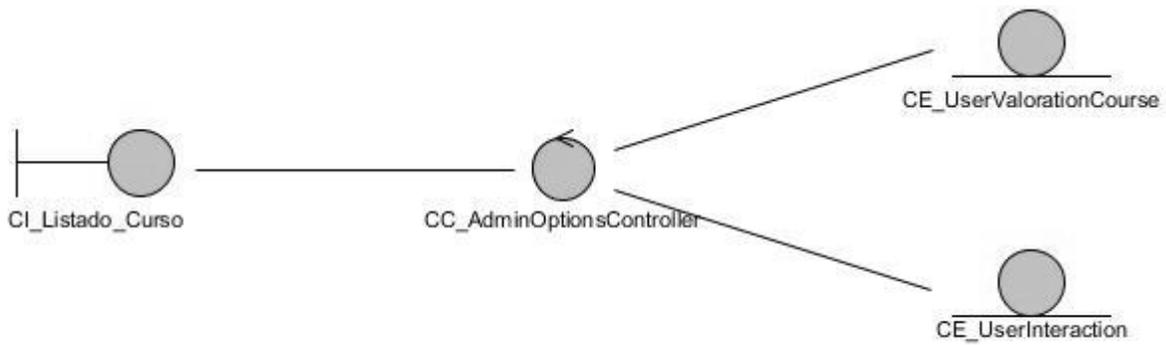


Diagrama de clases del análisis RF2

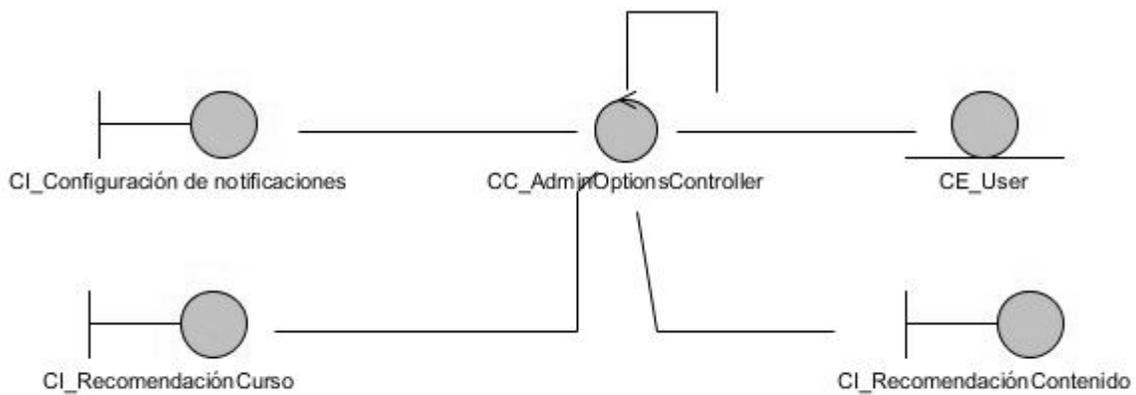


Diagrama de clases del análisis RF3

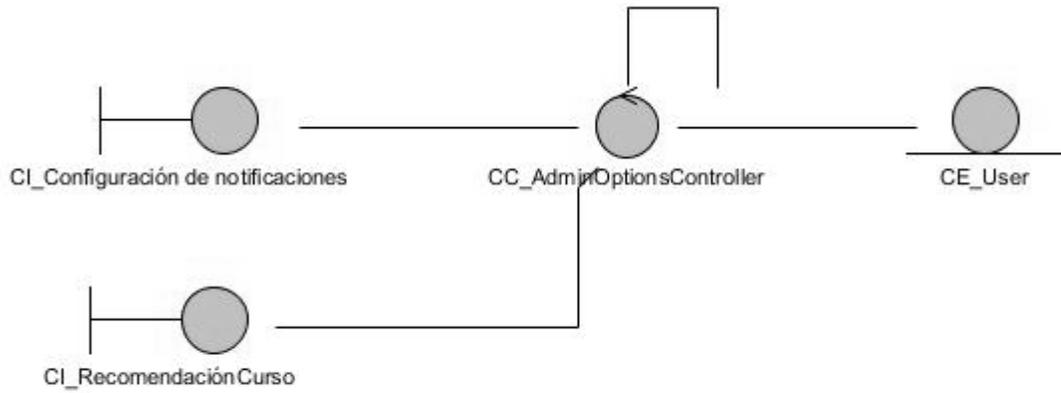


Diagrama de clases del análisis RF4

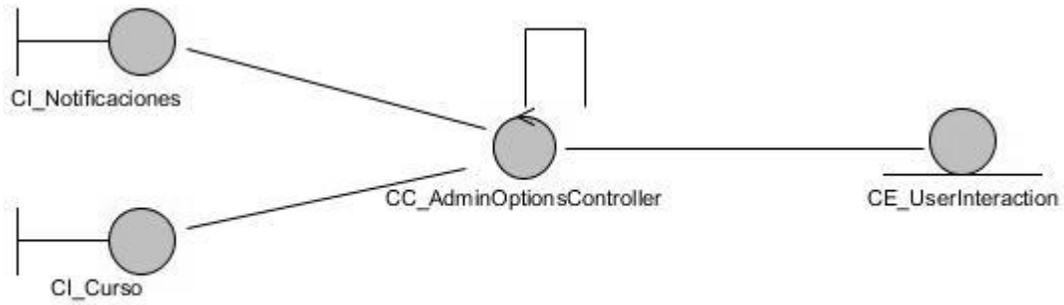


Diagrama de clases del análisis RF5

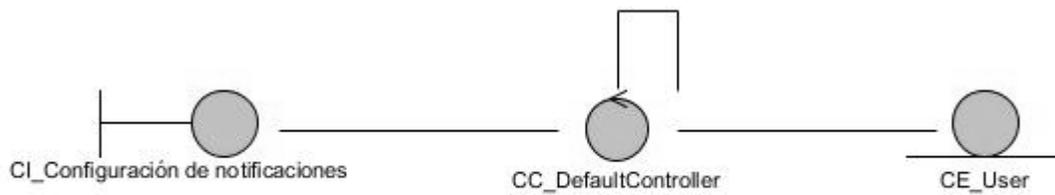


Diagrama de clases del análisis RF6

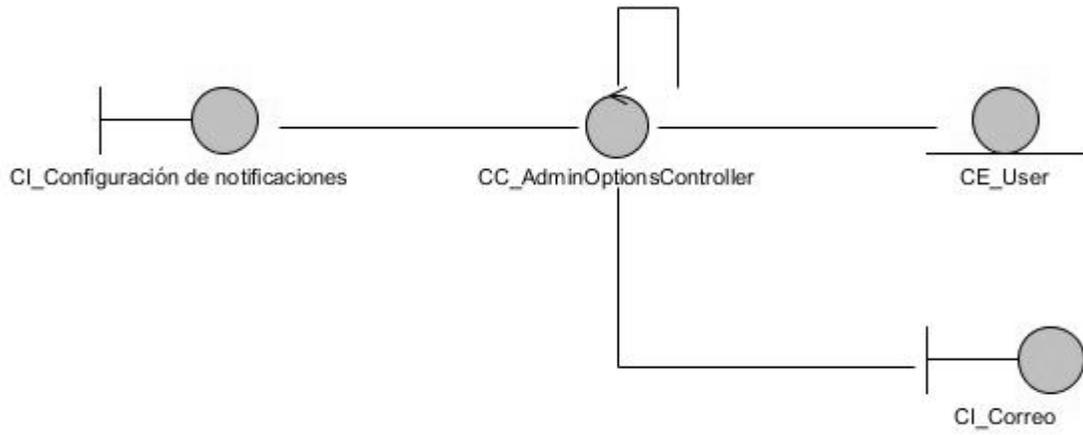


Diagrama de clases del análisis RF7.

Anexo 3: Diagramas de colaboración del análisis

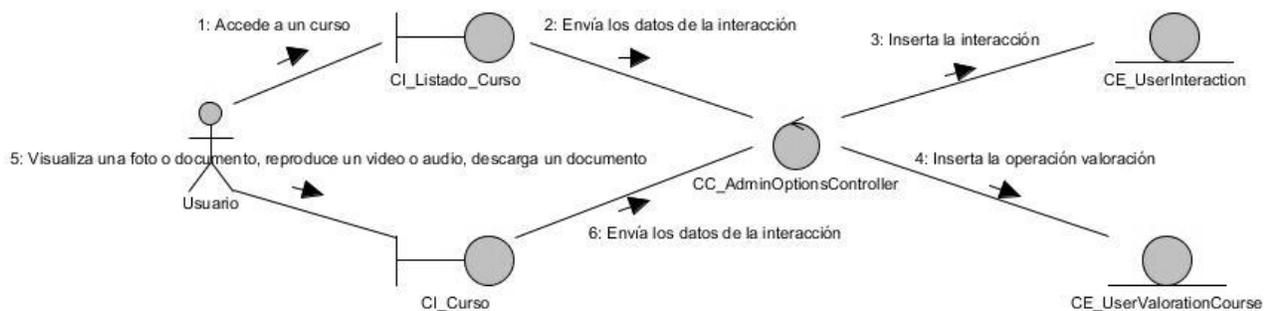


Diagrama de colaboración del análisis RF1

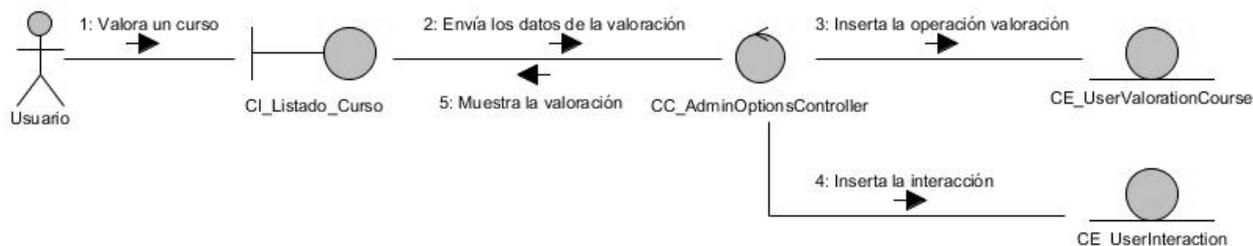


Diagrama de colaboración del análisis RF2

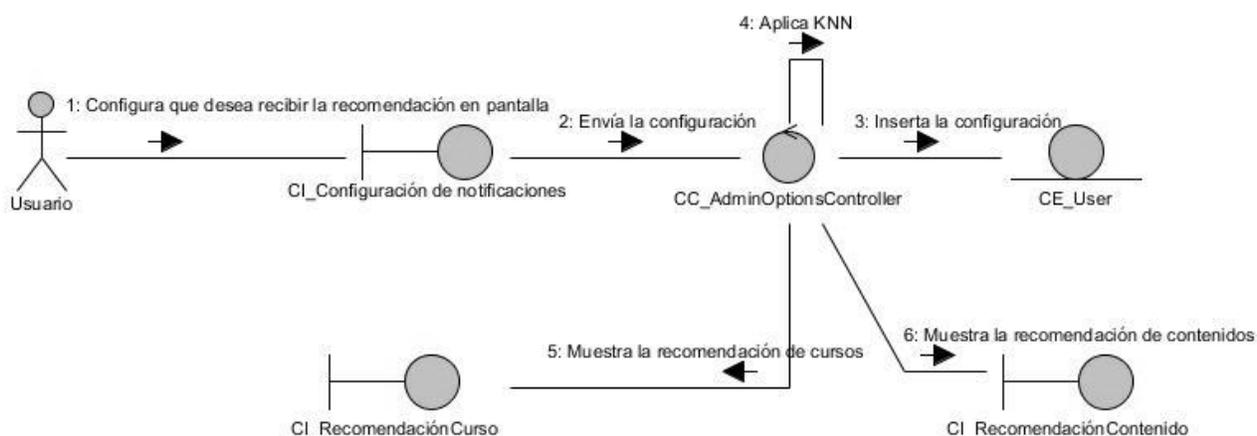


Diagrama de colaboración del análisis RF3

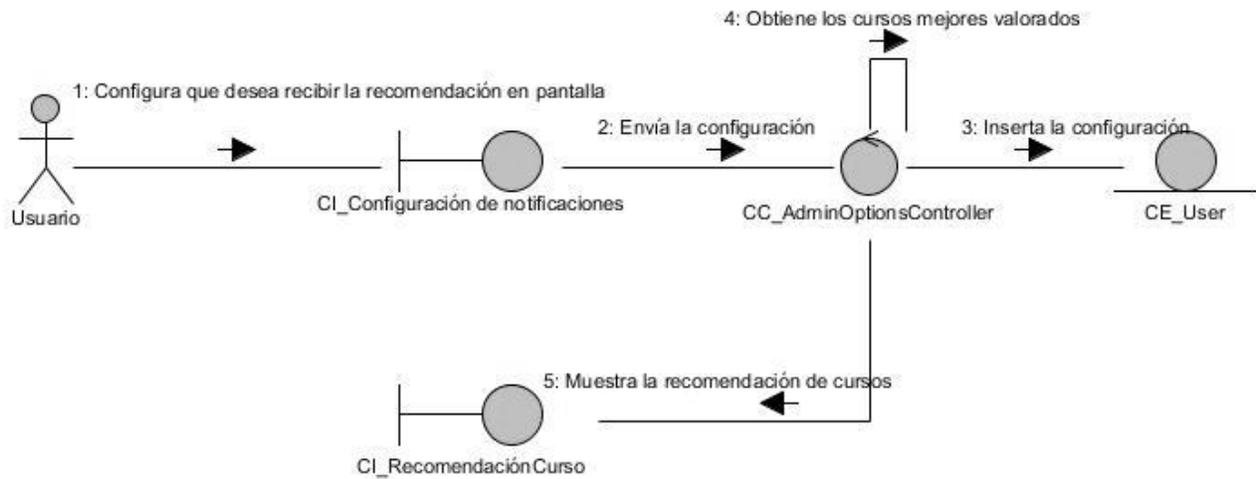


Diagrama de colaboración del análisis RF4

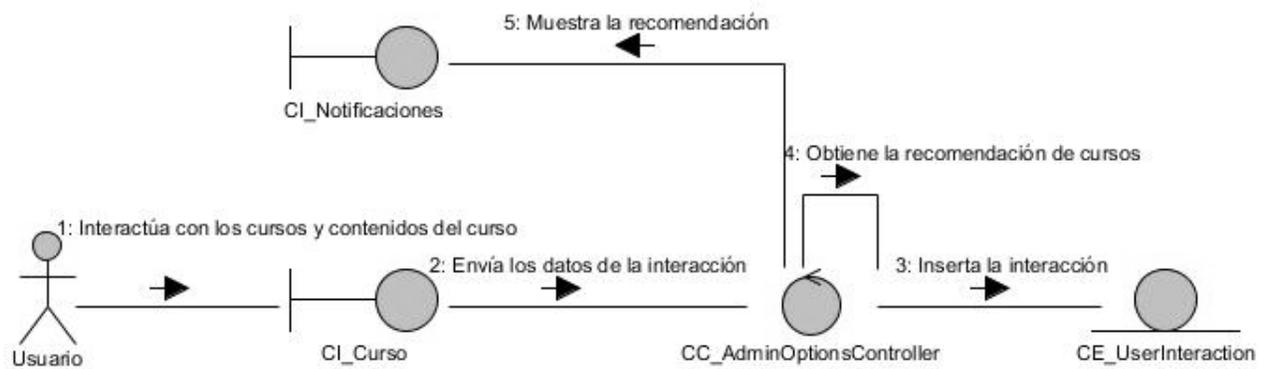


Diagrama de colaboración del análisis RF5

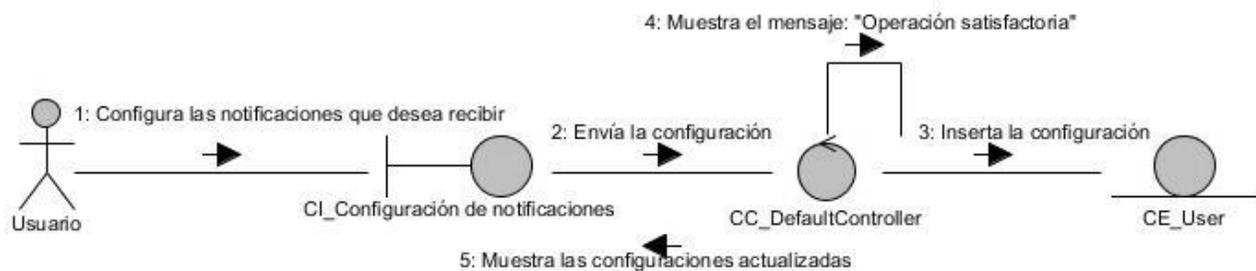


Diagrama de colaboración del análisis RF6

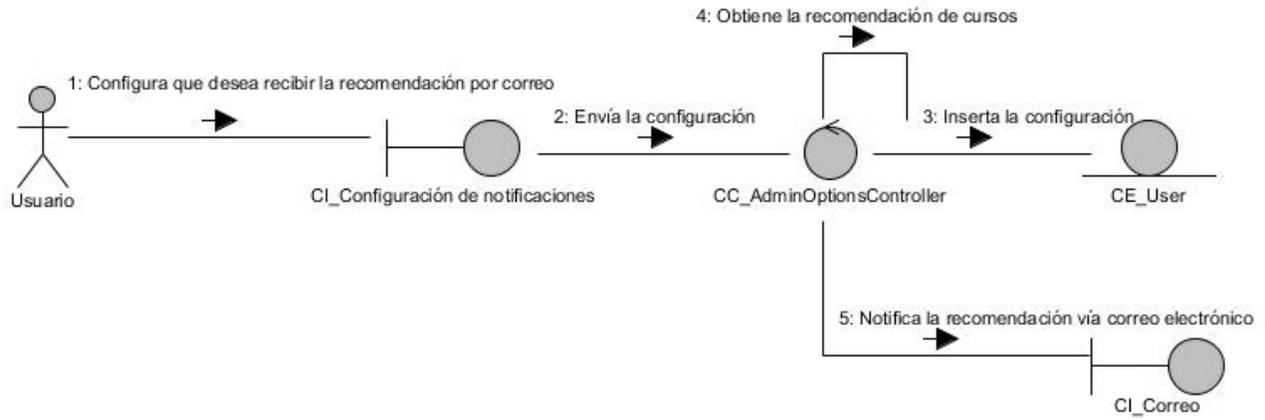


Diagrama de colaboración del análisis RF7

Anexo 4: Diagramas de clases del diseño

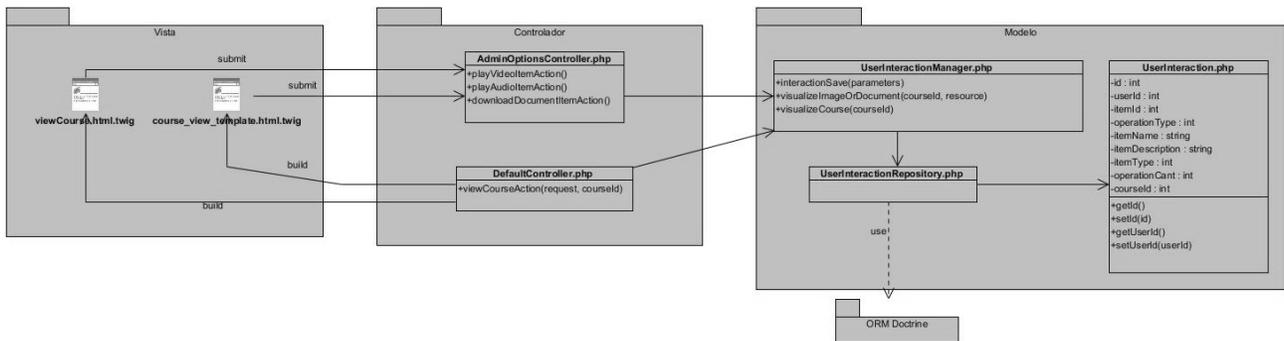


Diagrama de clases del diseño RF1

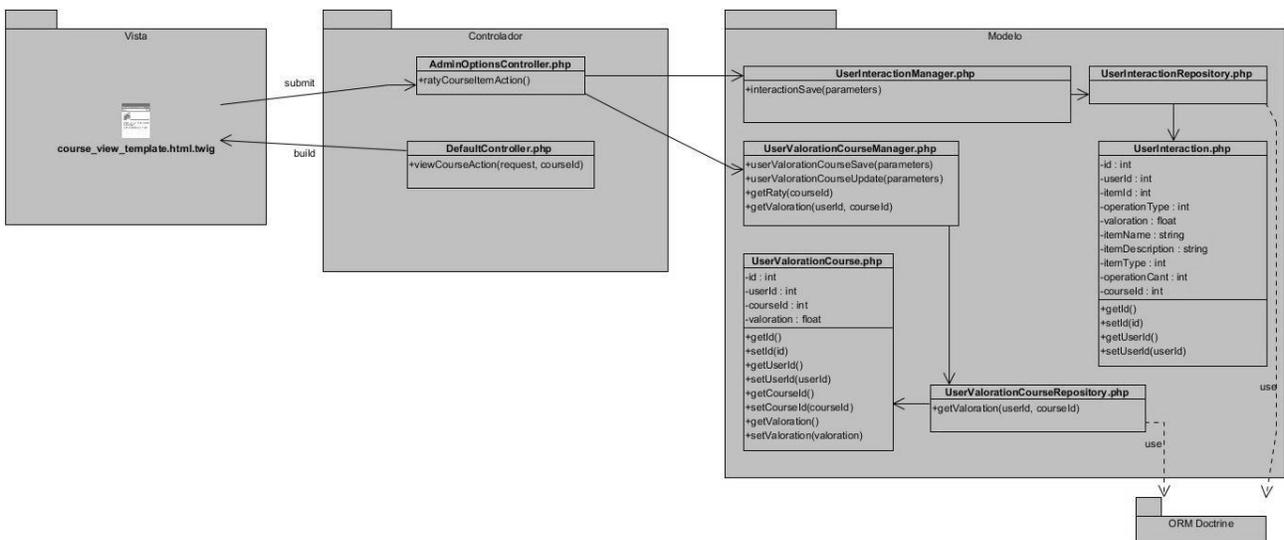


Diagrama de clases del diseño RF2

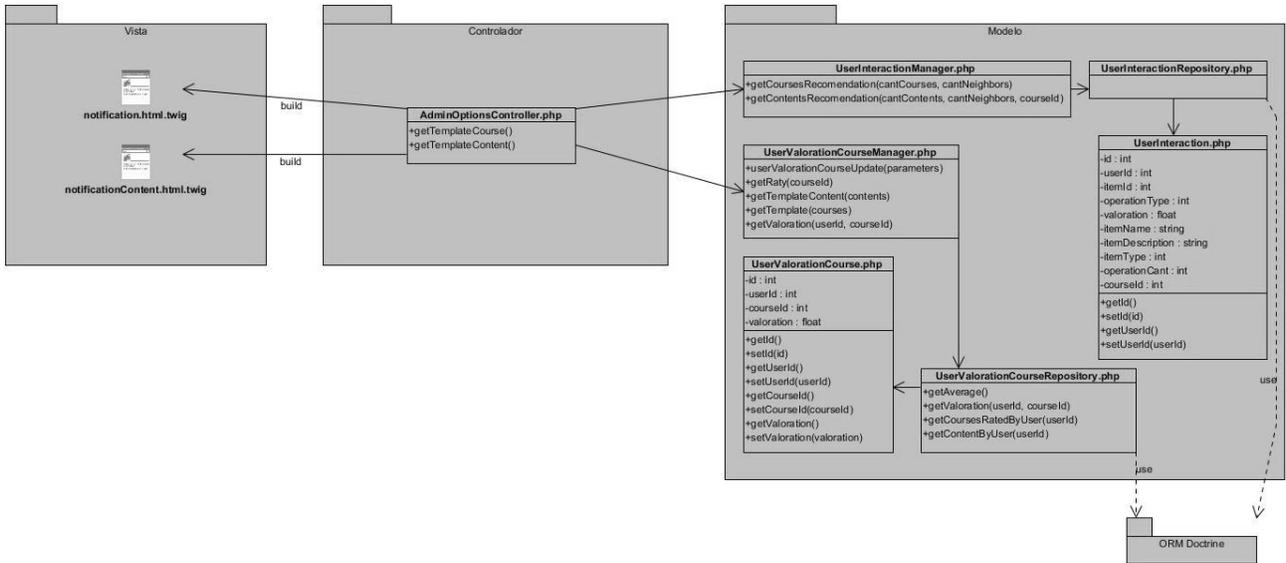


Diagrama de clases del diseño RF3

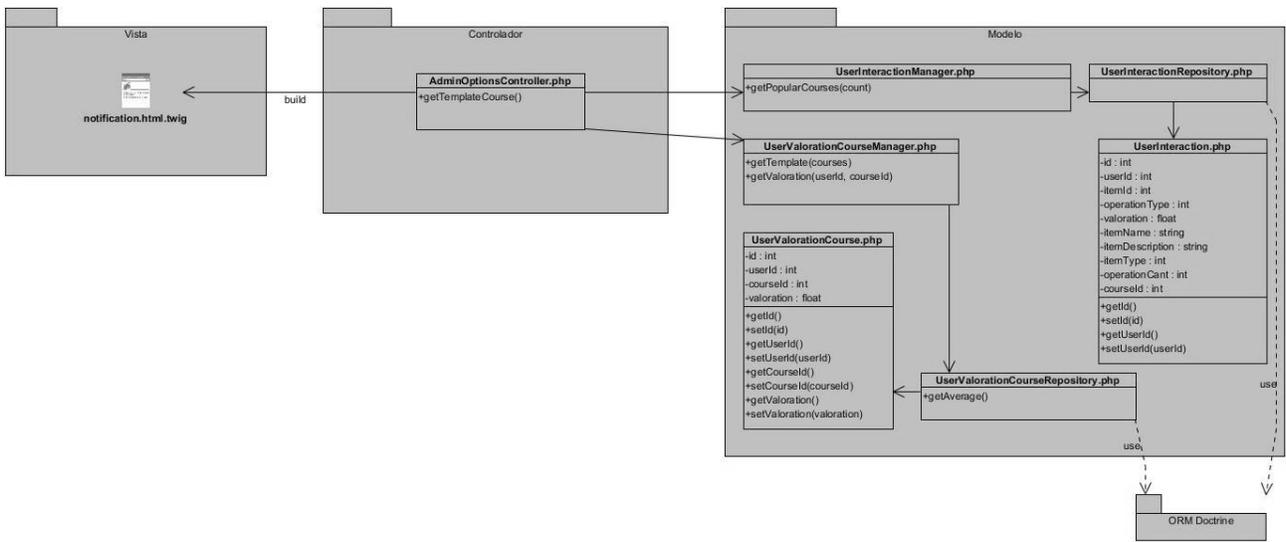


Diagrama de clases del diseño RF4

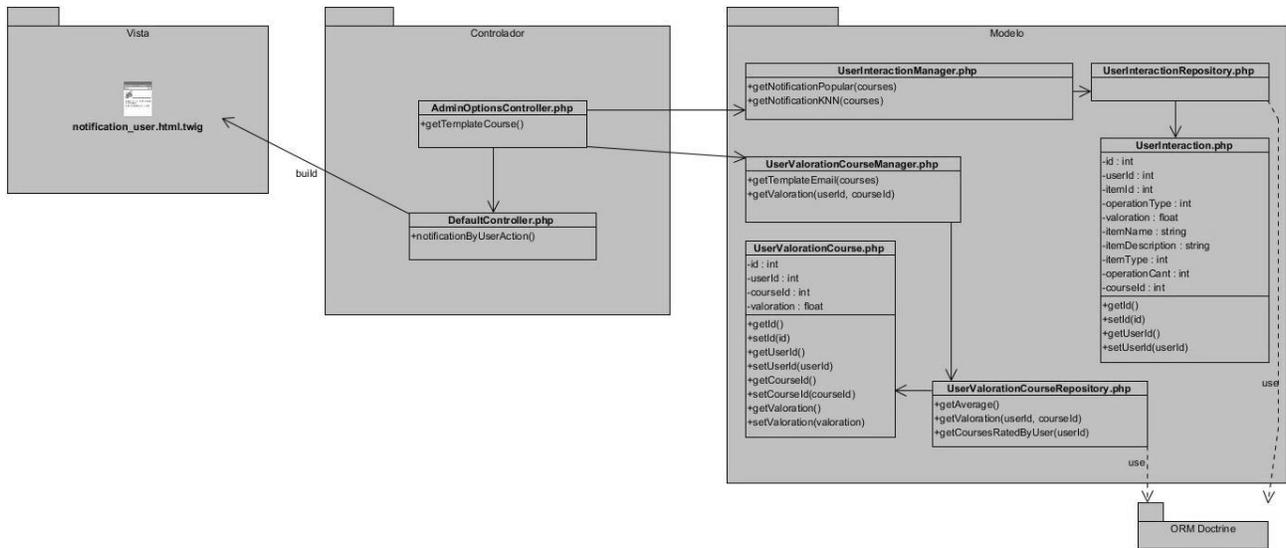


Diagrama de clases del diseño RF5

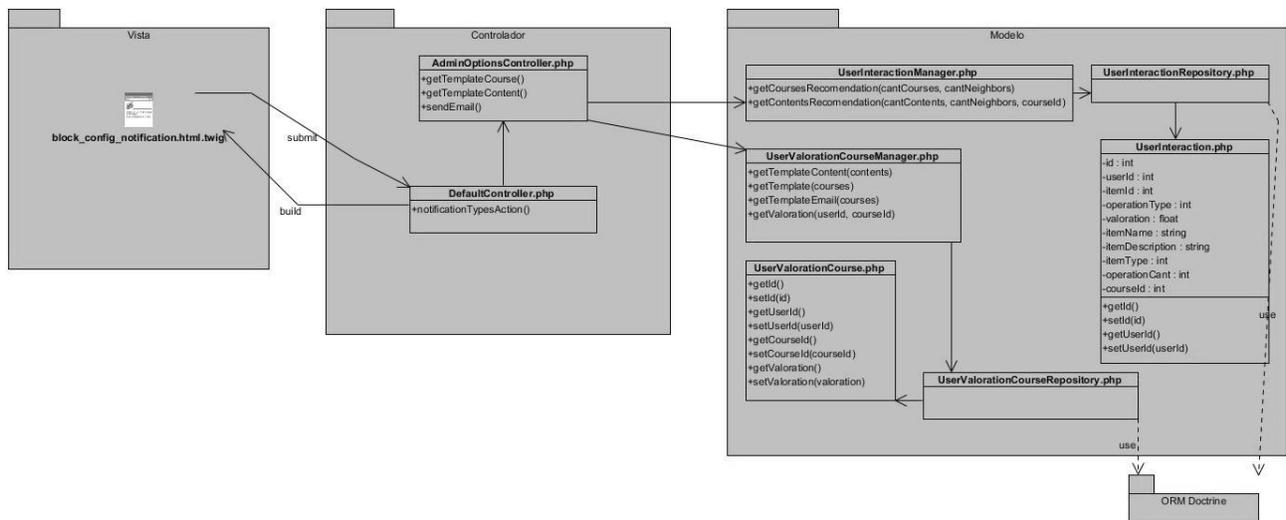


Diagrama de clases del diseño RF6

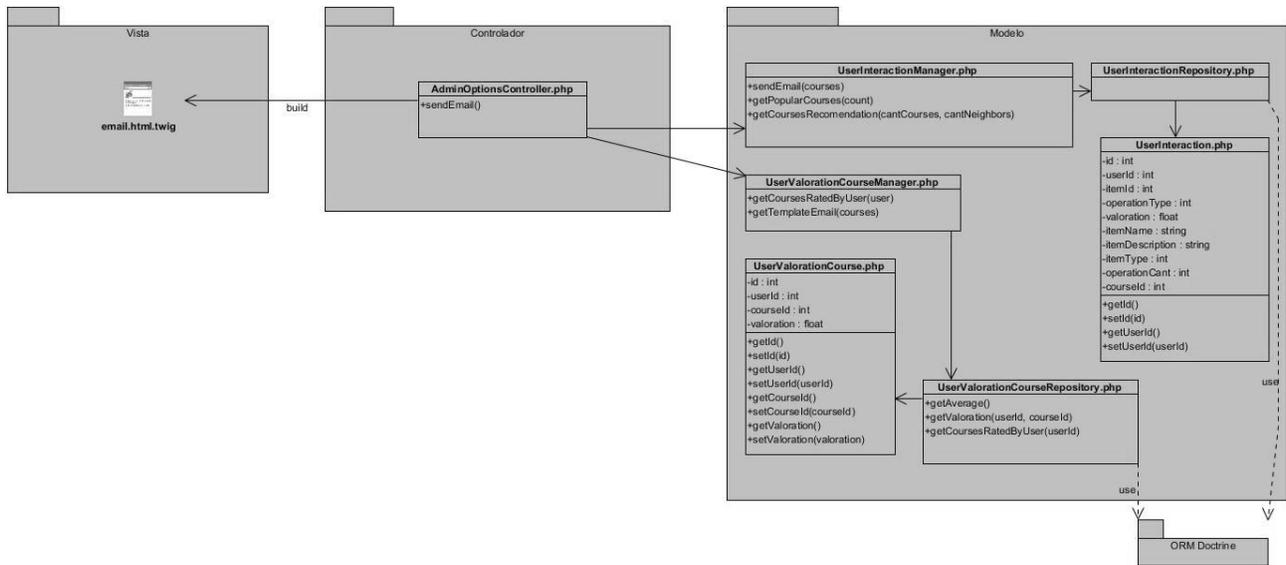


Diagrama de clases del diseño RF7

Anexo 5: Diagramas de secuencia del diseño

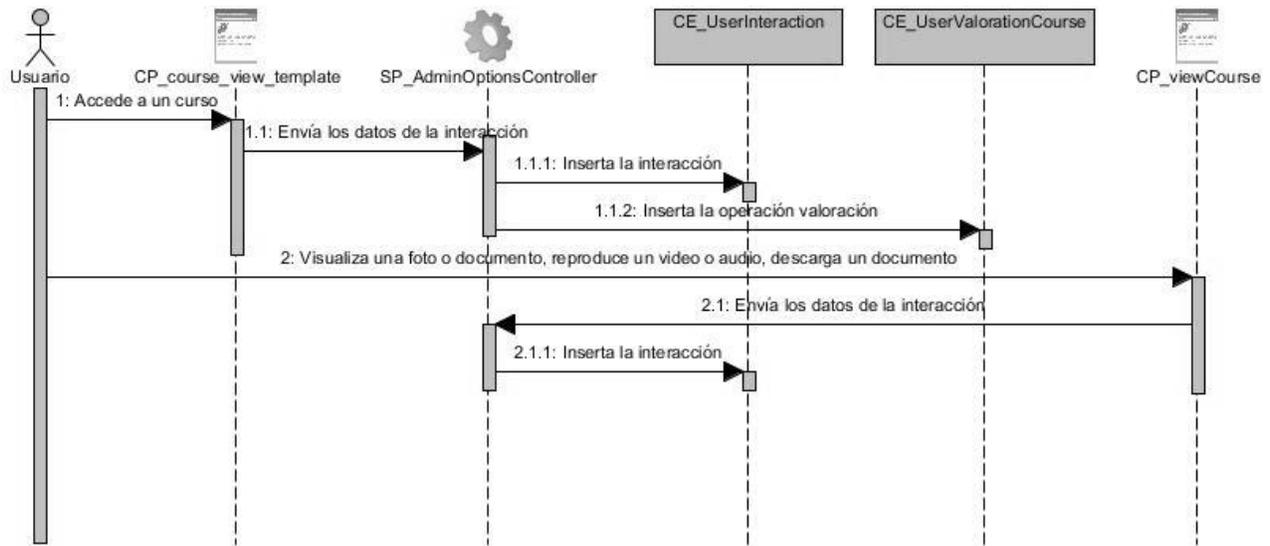


Diagrama de secuencia RF1

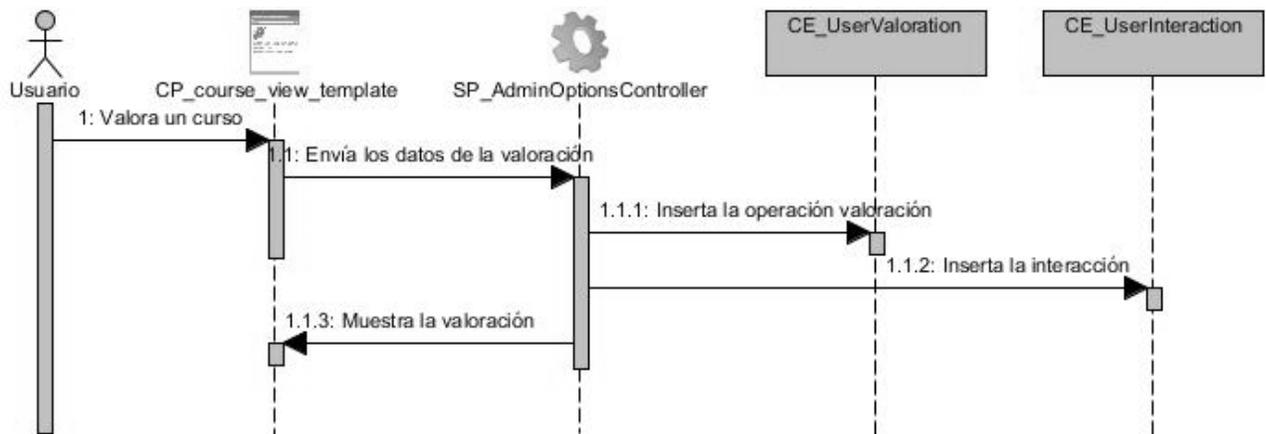


Diagrama de secuencia RF2

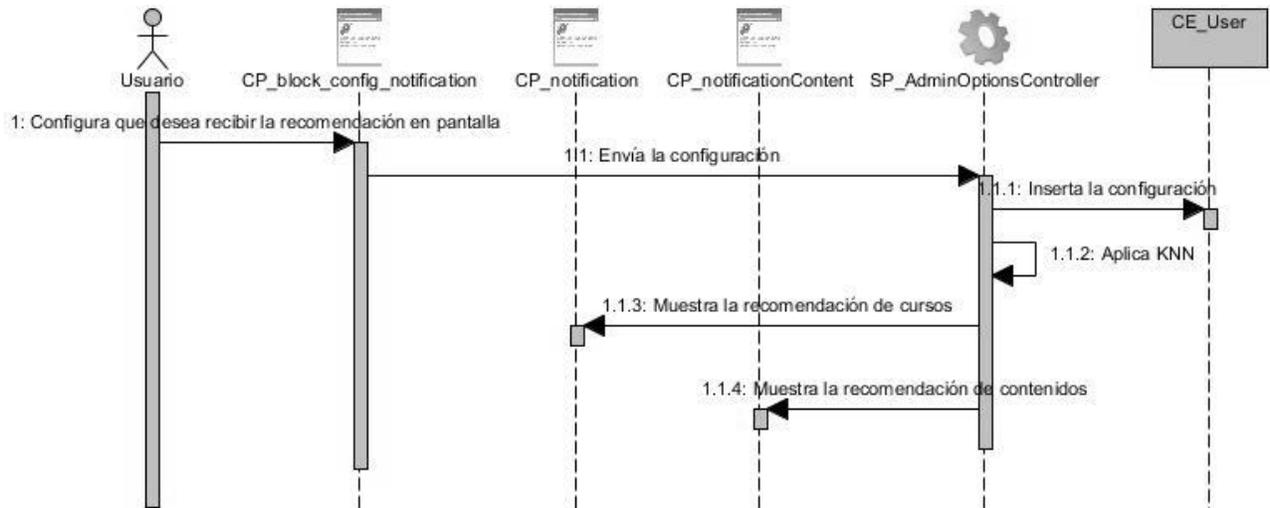


Diagrama de secuencia RF3

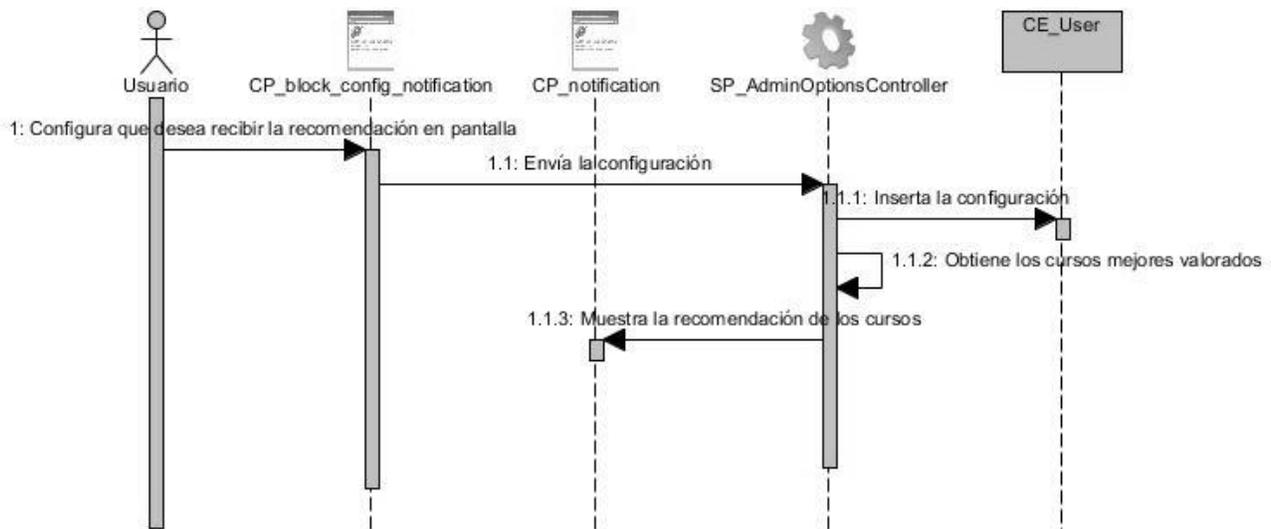


Diagrama de secuencia RF4

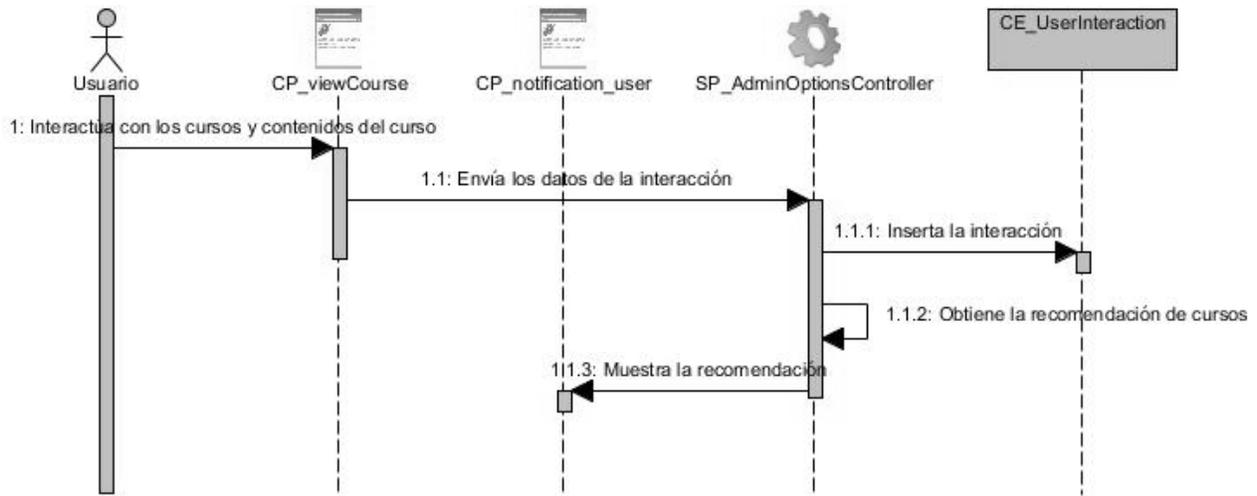


Diagrama de secuencia RF5

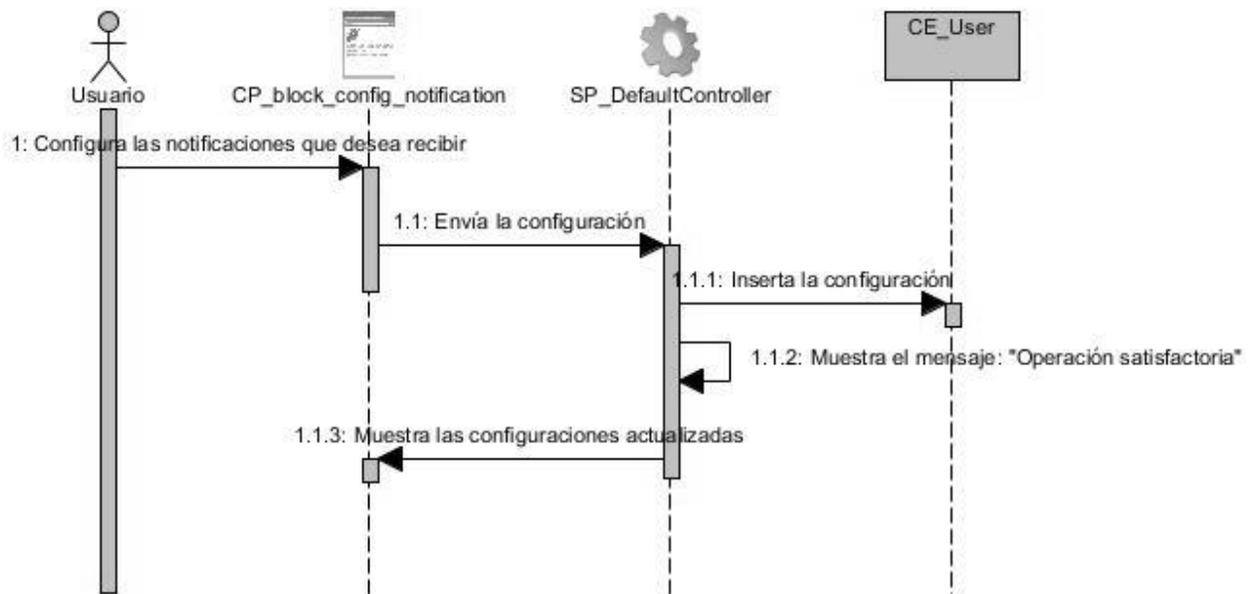


Diagrama de secuencia RF6

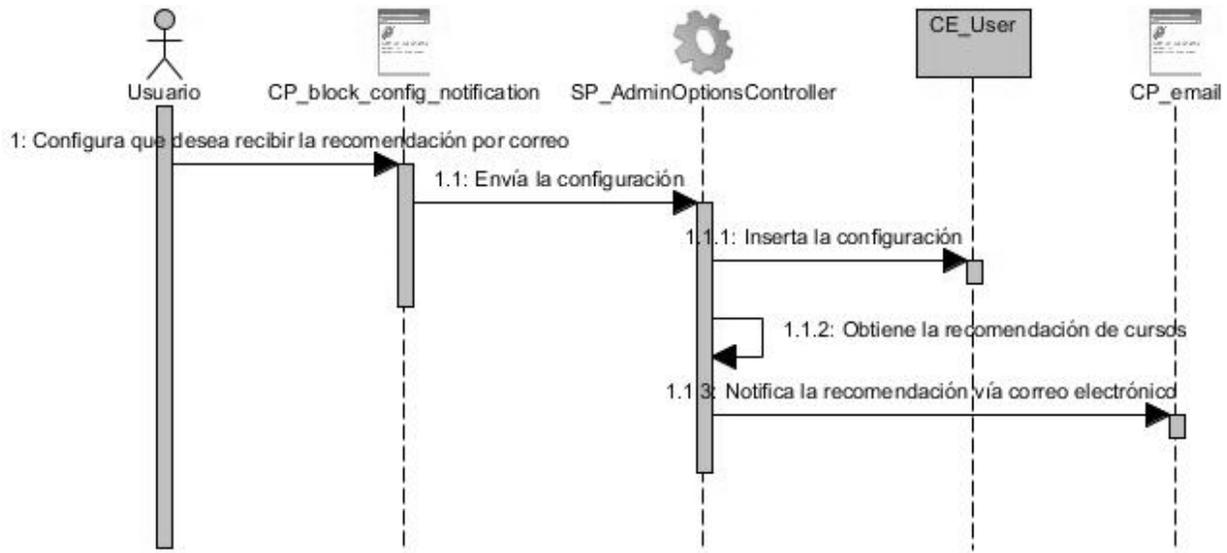


Diagrama de secuencia RF7

Anexo 6: Descripciones de las tablas de la base de datos

Descripción de la entidad "UserInteraction" de la base de datos

UserInteraction		
Descripción: En la presente tabla se muestran los atributos de la entidad UserInteraction.		
Atributo	Tipo	Descripción
id	integer(10)	Etiqueta única que identifica la interacción.
userId	integer(10)	Almacena el identificador del usuario que realizó la interacción.
itemId	integer(10)	Almacena el identificador del ítem interactuado por el usuario.
operationType	varchar(255)	Almacena el tipo de operación que hizo el usuario.
valuation	float(8)	Almacena la valoración de un usuario a un curso.
itemName	varchar(255)	Almacena el nombre del ítem interactuado por el usuario.
itemDescription	varchar(255)	Almacena la descripción del ítem interactuado por el usuario.
itemType	varchar(255)	Almacena el tipo de ítem interactuado por el usuario.
operationCant	integer(10)	Almacena la cantidad de veces que un usuario hizo determinada operación.
createdAt	timestamp	Almacena la fecha de la interacción.
updatedAt	timestamp	Actualiza la fecha de la interacción.

Descripción de la entidad "UserValorationCourse" de la base de datos

UserValorationCourse		
Descripción: En la presente tabla se muestran los atributos de la entidad UserValorationCourse.		
Atributo	Tipo	Descripción
id	integer(10)	Etiqueta única que identifica la valoración.
userId	integer(10)	Almacena el identificador del usuario que realizó la valoración.
valuation	float(8)	Almacena la valoración de un usuario a un curso.
courseId	integer(10)	Almacena el id del curso valorado por un usuario.
createdAt	timestamp	Almacena la fecha cuando el usuario valora un curso.
updatedAt	timestamp	Almacena la fecha actualizada cuando el usuario revalora un curso.

Descripción de la entidad "Resource" de la base de datos

Resource		
Descripción: En la presente tabla se muestran los atributos de la entidad Resource.		
Atributo	Tipo	Descripción
id	integer(8)	Etiqueta única que identifica el recurso.
name	varchar(255)	Almacena el nombre del recurso.
description	varchar(255)	Almacena la descripción del recurso.
file	varchar(255)	Almacena el archivo del recurso.
resourceType	integer(8)	Almacena el tipo de recurso.
source	varchar(255)	Almacena el origen del recurso.
metadata	varchar(255)	Almacena los metadatos del recurso.
isDownloadable	blob	Almacena si el recurso es descargable o no.
is_shared	blob	Almacena si el recurso es compartido o no.
author	varchar(255)	Almacena el autor del recurso.
created_at	timestamp	Almacena la fecha de creación del recurso.
updated_at	timestamp	Actualiza la fecha del recurso.

Descripción de la entidad "Course" de la base de datos

Course		
Descripción: En la presente tabla se muestran los atributos de la entidad Course.		
Atributo	Tipo	Descripción
id	integer(8)	Etiqueta única que identifica el curso.
name	varchar(255)	Almacena el nombre del curso.
shortName	varchar(255)	Almacena el nombre corto del curso.
summary	varchar(255)	Almacena la descripción del curso.
startDate	date	Almacena la fecha de inicio del curso.
closingDate	date	Almacena la fecha de fin del curso.
accessEnd	date	Almacena la fecha de cierre del curso.
image	varchar(255)	Almacena la dirección de la ubicación de la imagen del curso.
imageFile	clob	Almacena el archivo imagen del curso.
isShare	blob	Almacena si el curso es compartido o no.
isVisible	blob	Almacena si el curso es visible o no.
edition	integer(10)	Almacena la edición del curso.
configuration	varchar(255)	Almacena la configuración del curso.
createdAt	timestamp	Almacena la fecha de creación del curso.
updatedAt	timestamp	Actualiza la fecha del curso.

Descripción de la entidad "User" de la base de datos

User		
Descripción: En la presente tabla se muestran los atributos de la entidad User.		
Atributo	Tipo	Descripción
id	integer(8)	Etiqueta única que identifica el usuario.
name	varchar(255)	Almacena el nombre del usuario.
lastName	varchar(255)	Almacena el apellido del usuario.
githubId	varchar(255)	Almacena el id del usuario de ZERA en gitHub.
facebookId	varchar(255)	Almacena el id del usuario de ZERA en Facebook.
googleId	varchar(255)	Almacena el id del usuario de ZERA en Google.
spotifyId	varchar(255)	Almacena el id del usuario de ZERA en Spotify.
dropboxId	varchar(255)	Almacena el id del usuario de ZERA en Dropbox.
twitterId	varchar(255)	Almacena el id del usuario de ZERA en Twitter.
yahooId	varchar(255)	Almacena el id del usuario de ZERA en Yahoo.
amazonId	varchar(255)	Almacena el id del usuario de ZERA en Amazon.
instagramId	varchar(255)	Almacena el id del usuario de ZERA en Instagram.
config	varchar(255)	Almacena las configuraciones relacionadas al usuario.
enrollment	integer(8)	Almacena el rol del usuario.
created_at	timestamp	Almacena la fecha de creación del usuario.
updated_at	timestamp	Actualiza la fecha del usuario.

Anexo 7: Diseño de casos de prueba

Diseño de caso de prueba RF1

Escenario	Descripción	Datos de entrada	Respuesta del sistema	Flujo central
EC 1.1 Acceder a un curso.	Accede a un curso.	N/A	Guarda en base de datos los elementos de la interacción (idUserio, fecha, idCurso, nombreCurso, descripciónCurso, cantVeces, tipoDeltem y tipoOperación).	Página Principal/ Acceder a curso
EC 1.2 Visualizar foto.	Visualiza una foto.	N/A	Guarda en base de datos los elementos de la interacción (idUserio, fecha, idFoto, nombreFoto, descripciónFoto, cantVeces, tipoDeltem y tipoOperación).	Página Principal/ Acceder a curso/ Visualizar foto
EC 1.3 Visualizar documento.	Visualiza un documento.	N/A	Guarda en base de datos los elementos de la interacción (idUserio, fecha, idDocumento, nombreDocumento, descripciónDocumento, cantVeces, tipoDeltem y tipoOperación).	Página Principal/ Acceder a curso/ Visualizar documento
EC 1.4 Reproducir audio.	Reproduce un audio.	N/A	Guarda en base de datos los elementos de la interacción (idUserio, fecha, idAudio, nombreAudio, descripciónAudio, cantVeces, tipoDeltem y tipoOperación).	Página Principal/ Acceder a curso/ Reproducir audio
EC 1.5 Reproducir video.	Reproduce un video.	N/A	Guarda en base de datos los elementos de la interacción (idUserio, fecha, idVideo, nombreVideo, descripciónVideo, cantVeces, tipoDeltem y tipoOperación).	Página Principal/ Acceder a curso/ Reproducir video
EC 1.6 Descargar documento.	Descarga un documento.	N/A	Guarda en base de datos los elementos de la interacción (idUserio, fecha, idDocumento, nombreDocumento, descripciónDocumento, cantVeces, tipoDeltem y tipoOperación).	Página Principal/ Acceder a curso/ Descargar documento

Diseño de caso de prueba RF2

Escenario	Descripción	Datos de entrada	Respuesta del sistema	Flujo central
EC 1.1 Valorar un curso.	Valora un curso.	N/A	Muestra la valoración hecha por el usuario, guarda en base de datos la interacción.	Página Principal/ Valorar un curso

Diseño de caso de prueba RF3

Escenario	Descripción	Datos de entrada	Respuesta del sistema	Flujo central
EC 1.1 Mostrar recomendación de cursos obtenidos a partir del KNN en pantalla.	Muestra la recomendación de cursos obtenidos al aplicar KNN en pantalla.	N/A	Muestra los cursos recomendados por el algoritmo KNN en pantalla, el usuario tiene la opción de ir al curso de forma rápida haciendo click en el enlace. También puede elegir dejar de recibir este tipo de notificaciones.	Página Principal/ Acceder a curso
EC 1.2 Mostrar recomendación en pantalla con los contenidos obtenidos a partir del KNN.	Muestra la recomendación de los contenidos obtenidos al aplicar KNN en pantalla.	N/A	Muestra los contenidos recomendados por el algoritmo KNN en pantalla, el usuario puede elegir dejar de recibir este tipo de notificaciones.	Página Principal/ Acceder a curso

Diseño de caso de prueba RF4

Escenario	Descripción	Datos de entrada	Respuesta del sistema	Flujo central
EC 1.1 Mostrar recomendación en pantalla con los cursos mejores valorados.	Muestra en pantalla la recomendación de los cursos mejores valorados.	N/A	Muestra los cursos mejores valorados (los más populares) en pantalla, el usuario tiene la opción ir al curso de forma rápida haciendo click en el enlace. También puede elegir dejar de recibir este tipo de notificaciones.	Página Principal/ Acceder a curso

Diseño de caso de prueba RF5

Escenario	Descripción	Datos de entrada	Respuesta del sistema	Flujo central
EC 1.1 Visualizar recomendación de cursos obtenidos a partir de KNN.	Visualiza la recomendación de los cursos obtenidos al aplicar KNN.	N/A	Muestra la recomendación de los cursos obtenidos al aplicar KNN.	Notificaciones/ Ver todas
EC 1.2 Visualizar recomendación de los cursos mejores valorados.	Visualiza la recomendación de los cursos mejores valorados de la plataforma.	N/A	Muestra la recomendación de los cursos mejores valorados (más populares) de la plataforma.	Notificaciones/ Ver todas

Diseño de caso de prueba RF6

Escenario	Descripción	Datos de entrada	Respuesta del sistema	Flujo central
EC 1.1 Opción configurar notificaciones de tipo recomendación.	Selecciona la opción Configurar Notificaciones.	N/A	Brinda la posibilidad al usuario de configurar las notificaciones de tipo recomendación. Puede seleccionar recibir recomendación en pantalla, mediante correo electrónico, ambas o ninguna.	Perfil de usuario/ Notificaciones/ Configurar notificaciones
EC 1.2 Opción recibir recomendación en pantalla.	Selecciona la opción Plataforma y hace click en el botón Actualizar.	N/A	Actualiza el archivo json del perfil de usuario con las nuevas preferencias, muestra el mensaje: "Se ha realizado la acción satisfactoriamente". El usuario solo desea recibir notificaciones de tipo recomendación en pantalla.	Perfil de usuario/ Notificaciones/ Configurar notificaciones/ Actualizar
EC 1.3 Opción recibir recomendación vía correo electrónico.	Selecciona la opción Correo electrónico y hace click en el botón Actualizar.	N/A	Actualiza el archivo json del perfil de usuario con las nuevas preferencias, muestra el mensaje: "Se ha realizado la acción satisfactoriamente". El usuario solo desea recibir notificaciones de tipo recomendación vía correo electrónico.	Perfil de usuario/ Notificaciones/ Configurar notificaciones/ Actualizar
EC 1.4 Marcar ambas opciones.	Selecciona la opción Correo electrónico, la opción Plataforma y hace click en el botón Actualizar.	N/A	Actualiza el archivo json del perfil de usuario con las nuevas preferencias, muestra el mensaje: "Se ha realizado la acción satisfactoriamente". El usuario podrá recibir notificaciones de tipo recomendación por ambas vías.	Perfil de usuario/ Notificaciones/ Configurar notificaciones/ Actualizar
EC 1.5 Dejar de marcar ambas opciones.	No selecciona ninguna opción y hace click en el botón Actualizar.	N/A	Muestra un mensaje de error con la siguiente información: "Seleccione Estrategia para Almacenar". Regresa al EC 1.1	Perfil de usuario/ Notificaciones/ Configurar notificaciones/ Actualizar

Diseño de caso de prueba RF7

Escenario	Descripción	Datos de entrada	Respuesta del sistema	Flujo central
EC 1.1 Notificar recomendación de los cursos obtenidos al aplicar KNN vía correo electrónico.	Notifica vía correo electrónico los cursos obtenidos a partir de KNN.	N/A	Envía un correo electrónico con los cursos obtenidos al aplicar KNN. El usuario puede ir de forma rápida al curso haciendo click en el enlace.	Página Principal/ Acceder a curso
EC 1.2 Notificar recomendación de los cursos mejores valorados vía correo electrónico.	Notifica vía correo electrónico los cursos mejores valorados.	N/A	Envía un correo electrónico con los cursos mejores valorados (los más populares) de la plataforma. El usuario puede ir de forma rápida al curso haciendo click en el enlace.	Página Principal/ Acceder a curso