



Universidad de las Ciencias Informáticas

Facultad 4

Sistema para recomendar patrones de diseño de recursos educativos basado en el tratamiento de agrupaciones y la calidad percibida por los usuarios

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

Autor:

Naylin Corrales Sánchez

Tutores:

Ing. Yasirys Terry González

Ing. Reiman Alfonso Azcuy

La Habana, junio, 2018



(...) nuestra juventud debe procurar adquirir aquellos conocimientos que sean más útiles en cada momento a la nación. Sobre todo, si se tiene en cuenta que estamos entrando en una etapa enteramente nueva (...)"

Universidad de la Habana, 11 de mayo de 1959

Fidel Castro Ruz

Declaración de autoría

Declaro que soy la única autora del presente trabajo y otorgo a la Universidad de las Ciencias Informáticas los derechos primordiales del mismo, con carácter exclusivo.

Para que así conste firmo la presente a los __ días del mes de _____ del año 2018.

Naylin Corrales Sánchez

Tutora: Ing. Yasirys Terry González

Tutor: Ing. Reiman Alfonso Azcuy

Dedicatoria

A mi abuelo Héctor Benítez Rubier.

A mis padres Dulce María y José Vidal.

A mi abuela Esperanza Nicolasa y mi hermana Naida Verónica.

Agradecimientos

A nuestro eterno Comandante en Jefe Fidel Castro Ruz por crear el "Proyecto Futuro".

A la Universidad de las Ciencias Informáticas por formarme como profesional.

A mi abuelo Héctor Benítez Rubier por ser mi motor impulsor.

A mi abuela por su dedicación, paciencia y cuidado en cada etapa de mi vida.

A mis padres por su amor incondicional y su entera confianza.

A mi hermana por hacerme reír en momentos de tristeza y ver la vida desde otro punto de vista.

A mis sobrinos por sus tiernas travesuras.

A mis tías y primos por su preocupación en todos estos años.

A David por su amor, su confianza y su infinita paciencia.

A la familia de David en especial a mis suegros y cuñado por acogerme como un miembro más de la familia.

A mis tutores Yasirys y Reiman por su tiempo, las noches de desvelo, su entrega y su sincera amistad.

A mi mejor amiga Yusleidy por estar presente en cada momento que la he necesitado y no dudar nunca.

A mis amigos Hermes, Carrión, Parker, Yoandi, Elianis, Luisi y Maydalis por compartir cada día una nueva experiencia.

A mis compañeras de apartamento Jessica, Inés, Taire, Caridad, Jeniffer y Adis por compartir momentos inolvidables.

A aquellas personas que de una forma u otra me ofrecieron su ayuda para la realizar este sueño realidad.

A los que me preguntaron ¿Cómo va la tesis?

Gracias todos.

Resumen

Con el auge de la utilización de las nuevas tecnologías en la educación superior cubana, ha aumentado progresivamente el uso de los recursos educativos. Al confeccionar un recurso educativo se emplean patrones de diseño, estos proporcionan herramientas necesarias para construirlos con una estructura consistente. Cada patrón de diseño resuelve por si solo un problema de diseño de un recurso educativo en particular, pero al relacionarlo con otros patrones, brindan una respuesta más amplia en cuanto al contexto dado. Por ello, estos patrones se agrupan fundamentalmente en lenguajes y catálogos. Actualmente, existe gran cantidad de patrones de diseño que pudieran utilizarse para resolver problemas de diseño de recursos educativos, ellos se encuentran almacenados en diversas fuentes. A los diseñadores de recursos educativos se les dificulta la decisión de cuáles patrones pudieran aplicar para resolver un problema determinado. Además, se pudiera conocer el nivel de aceptación que tienen los usuarios al interactuar con determinados patrones. Por esta razón, se propone el “Sistema para recomendar patrones de diseño de recursos educativos basado en el tratamiento de agrupaciones y la calidad percibida por los usuarios”, capaz de, a partir de la introducción del problema de diseño de recursos educativos, realizando una búsqueda federada en los distintos repositorios de patrones de diseño, analizando las agrupaciones de patrones de diseño y teniendo en cuenta el nivel de satisfacción de cada usuario al interactuar con determinado patrón, recomienda los adecuados atendiendo a sus necesidades.

Palabras claves: recurso educativo, patrones de diseño, lenguajes de patrones, catálogos de patrones, nivel de satisfacción.

Índice de contenido

Resumen	VI
Introducción	1
Capítulo 1. Marco teórico sobre los sistemas de recomendación y los patrones de diseño de recursos educativos	6
1.1 Recursos educativos	6
1.2 Patrones de diseño.....	7
1.3 Lenguajes de patrones de diseño	8
1.4 Catálogos de patrones de diseño	9
1.5 Sistemas de recomendación.....	10
1.6 Técnica de n-gramas contextuales	15
1.7 Búsquedas federadas.....	16
1.8 Análisis de soluciones similares.....	16
1.9 Herramientas y tecnologías a utilizar	19
1.10 Interoperabilidad.....	23
1.11 Conclusiones parciales.....	23
Capítulo 2. Sistema de recomendación de patrones de diseño de recursos educativos ..	24
2.1 Propuesta de solución	24
2.1.1 Comparación entre textos.....	27
2.1.2 Calidad percibida por los usuarios.....	27
2.2 Arquitectura	29

2.3	Conexión con repositorios	30
2.4	Patrones arquitectónicos	32
2.5	Requisitos funcionales	33
2.6	Requisitos no funcionales	34
2.7	Historias de usuario	36
2.8	Conclusiones parciales	47
Capítulo 3. Implementación y validación del sistema de recomendación		48
3.1	Implementación	48
3.1.1	Estándares de codificación	48
3.1.2	Funcionalidades implementadas.....	49
3.2	Pruebas	51
3.3	Técnica ladov	58
3.4	Conclusiones parciales	61
Conclusiones Generales.....		63
Recomendaciones		64
Referencias		65
Anexos		71

Índice de tablas

Tabla 1. Cuadro comparativo entre los sistemas similares.....	18
Tabla 2. HU Validar campo de texto contraseña.....	36
Tabla 3. HU Introducir el problema de recursos educativos.....	38
Tabla 4. HU Listar los patrones de diseño obtenidos en el análisis comparativo.....	40
Tabla 5. HU Listar los lenguajes de patrones de diseño obtenidos en el análisis comparativo.....	42
Tabla 6. HU Listar los catálogos de patrones de diseño.....	44
Tabla 7. Mostrar la procedencia de los patrones recomendados.....	45
Tabla 8. Descripción de variables.....	54
Tabla 9. Caso de prueba RF. Establecer la conexión con los repositorios de patrones de diseño.....	55
Tabla 10. Caso de prueba RF. Introducir el problema de diseño de recursos educativos	56
Tabla 11. Caso de prueba RF. Mostrar la procedencia de los patrones recomendados...	57
Tabla 12. Cuadro lógico de ladov.....	58
Tabla 13. Niveles de satisfacción.....	60
Tabla 14. Resultados de los encuestados.....	61

Índice de ilustraciones

Ilustración 1. Sistemas de recomendación colaborativa	11
Ilustración 2. Sistemas de recomendación basado en contenido	12
Ilustración 3. Sistemas de recomendación basado en conocimiento.....	13
Ilustración 4. Sistemas de recomendación híbrida	14
Ilustración 5. Diagrama de procesos del flujo de datos del sistema	25
Ilustración 6. Ecuación de medida de semejanza	27
Ilustración 7. Ecuación para determinar la calidad percibida por los usuarios	28
Ilustración 8. Código para establecer la conexión	49
Ilustración 9. Código para la similitud entre cadenas	50
Ilustración 10. Código para cuando las cadenas son semejantes	51
Ilustración 11. Prueba de Caja Blanca	53
Ilustración 12. Prueba de caja negra.....	53
Ilustración 13. Fórmula grupal de ladov	60

Introducción

Con el auge de la utilización de las “Tecnologías de la Información y de las Comunicaciones” (TICs), se ha facilitado el proceso de enseñanza - aprendizaje. Las TICs “ayudan a lograr el acceso universal a la educación, mejoran la igualdad y la calidad de la misma; también contribuyen al desarrollo profesional de los docentes, a la mejora de la gestión, la gobernanza y la administración de la educación, siempre y cuando se apliquen las políticas, las tecnologías y las capacidades adecuadas” [1].

Mediante la *World Wide Web* (www) se puede acceder a un conjunto inmenso de páginas web, ubicadas en servidores de todo el mundo, que están conectados entre sí mediante la red Internet. El simple acceso a la información está derivando hacia procesos de comunicación más complejos y sofisticados con la incorporación de herramientas, que proporcionan nuevas e importantes perspectivas de futuro en términos generales y también para la educación.

Con el empleo de estas tecnologías ha aumentado progresivamente la creación de los recursos educativos abiertos, estos “son materiales de enseñanza, aprendizaje o investigación que se encuentran en el dominio público o que han sido publicados con una licencia de propiedad intelectual que permite su utilización, adaptación y distribución gratuita” [2].

Estos recursos educativos se pueden almacenar en diferentes repositorios a través de servicios prestados (...) para recopilar, gestionar, difundir y preservar la producción documental de una institución (...). Básicamente, los repositorios son sitios en los cuales se alojan objetos digitales como vídeos, animaciones, imágenes, documentos, libros, entre otras. Asimismo, son organizados por categorías, áreas de conocimiento, temas y tipo, por mencionar algunos. Su objetivo es facilitar la búsqueda, acceso y visualización del material digital, así como la preservación y difusión de materiales [3].

Para la confección de recursos educativos es necesario el empleo de patrones de diseño, pues estos proporcionan a los diseñadores crearlos con una mejor estructura. Existen características que se pueden apreciar como ventajas al utilizar los patrones de diseño de recursos educativos, una de ellas es la reusabilidad. Para que los patrones de diseño logren

una mayor reusabilidad pudieran aplicarse en diversas situaciones de aprendizaje y adaptarse a nuevas soluciones, modificando su contenido [4] [5].

Se define como patrón de diseño a una solución probada para un problema en un contexto en específico. Cada problema documenta una solución reutilizable, encapsula el conocimiento sobre las prácticas exitosas y proporciona información sobre su utilidad y sus compensaciones. Algunos patrones han sido catalogados en colecciones o bibliotecas de patrones. [6] [7] [8] [9].

Actualmente, existe gran cantidad de patrones de diseño almacenados en diferentes repositorios, que pudieran utilizarse para la confección de recursos educativos. Provocando que a los diseñadores de recursos educativos se les dificulte la obtención de estos patrones, la determinación de cuáles de ellos reutilizar para solucionar un problema de diseño dado y la creación de nuevos patrones, que en algunos casos son similares a los que ya existen.

A consecuencia de lo antes expuesto, en el curso 2014-2015, se desarrolló en la Facultad 4 de la Universidad de las Ciencias Informáticas (UCI) el Sistema de Recomendación de Patrones de Diseño de Recursos Educativos Abiertos (REPREA).

REPREA [10] es un sistema de recomendación basado en conocimiento, utilizando la información sobre un dominio para arribar a la solución de un problema. A través de una similitud de cadenas entre el problema introducido por los usuarios y la descripción del problema del patrón almacenado, recomienda una serie de patrones de diseño de recursos educativos disponibles en los repositorios. Para poder realizar la recomendación, el sistema establece una conexión a los distintos repositorios de patrones de diseño, realizando búsquedas federadas.

Los patrones de diseño recomendados son considerados como posibles soluciones, pero en ocasiones un solo patrón de diseño no brinda una respuesta a determinados problemas. Por esta razón surgen las agrupaciones de patrones de diseño, las cuales son los catálogos y los lenguajes. Estas agrupaciones se deben considerar para que sean analizadas primeramente antes que un patrón independiente. Las limitaciones detectadas en el sistema REPREA son las siguientes:

- No le brinda a los usuarios información de la procedencia del patrón recomendado: cuando se desea buscar información detallada de los patrones recomendados se desconoce la fuente donde adquirir dicha información.
- Realiza un análisis de la totalidad de los patrones de diseño almacenados: no se verifica si la solución del problema podría ser a través de un lenguaje o de patrones pertenecientes a un catálogo, realizando búsquedas innecesarias en los repositorios de patrones de diseño.
- No tiene presente la calidad percibida por los usuarios: se desconoce el nivel de satisfacción que tienen los usuarios al interactuar con determinados patrones de diseño. Si se toman en cuenta algunos de los parámetros que permiten medir este nivel de satisfacción se podrá recomendar patrones de diseño adecuados a las necesidades de los usuarios.

A partir de las limitaciones encontradas en REPREA se origina el siguiente **problema a resolver**: ¿Cómo recomendar patrones de diseño de recursos educativos adecuados a las necesidades de los usuarios?

Se define como **objeto de estudio**: la recomendación de patrones de diseño de recursos educativos, siendo el **campo de acción**: la recomendación de patrones de diseño basado en el tratamiento de agrupaciones y la calidad percibida por los usuarios.

El **objetivo general** de la investigación es: desarrollar el Sistema para recomendar patrones de diseño de recursos educativos basado en el tratamiento de agrupaciones y la calidad percibida por los usuarios para solucionar problemas de diseño de recursos educativos.

Como **objetivos específicos** se determinaron los siguientes:

1. Fundamentar los conceptos, características y antecedentes de los sistemas de recomendación mediante un estudio bibliográfico.
2. Diseñar el Sistema para recomendar patrones de diseño de recursos educativos basado en el tratamiento de agrupaciones y la calidad percibida por los usuarios.

3. Implementar el Sistema para recomendar patrones de diseño de recursos educativos basado en el tratamiento de agrupaciones y la calidad percibida por los usuarios.
4. Validar el Sistema para recomendar patrones de diseño de recursos educativos basado en el tratamiento de agrupaciones y la calidad percibida por los usuarios.

Para dar cumplimiento al objetivo propuesto se tiene como **hipótesis** que, si se desarrolla un Sistema para recomendar patrones de diseño de recursos educativos basado en el tratamiento de agrupaciones y la calidad percibida por los usuarios, se contribuirá a la selección de patrones adecuados para solucionar problemas de diseño de recursos educativos.

Métodos teóricos

Para la realización de esta investigación se utilizarán métodos que permiten descubrir en el objeto de estudio las relaciones esenciales y las cualidades fundamentales. Por ello, el presente trabajo se apoya básicamente en los procesos de análisis y síntesis.

Los métodos que serán empleados son:

- **Histórico-Lógico:** este método será usado para hacer un estudio acerca del origen de los sistemas de recomendación y su evolución hasta la actualidad. Este estudio permitirá adquirir referentes teóricos e históricos para ver cómo se han ido perfeccionando a través del tiempo y sirva como base para el desarrollo del sistema como resultado de la presente investigación.
- **Analítico-Sintético:** este método será utilizado para analizar y resumir la información adquirida de bibliografías y materiales sobre el proceso de recomendación y los patrones de diseño de recursos educativos. Permitiendo lograr un mayor entendimiento acerca del funcionamiento de los sistemas de recomendación y conocer mejor sus principales características, para escoger las que tendrá el sistema a desarrollar como resultado de la presente investigación.
- **Encuesta:** realizada entre diseñadores de recursos educativos de la universidad para la validación del sistema.

- **Entrevista:** la entrevista utilizada es la no estructurada. Se entrevistó al cliente para conocer las funcionalidades que se deben incorporar al sistema.

El presente trabajo está estructurado de la siguiente manera: introducción, tres capítulos, conclusiones, referencias, recomendaciones y anexos.

Capítulo 1: Marco teórico los sistemas de recomendación y los patrones de diseño de recursos educativos

Se definen los conceptos de recursos educativos, patrones de diseño, lenguajes de patrones, patrones pertenecientes a catálogos, así como la evolución de los sistemas de recomendación, sus características y los tipos de sistemas de recomendación que existen. También se analizan las herramientas y tecnologías a utilizar para el desarrollo del Sistema de Recomendación.

Capítulo 2: Sistema de recomendación de patrones de diseño de recursos educativos

Se describe la arquitectura, patrones arquitectónicos y patrones de diseño que se emplearán para la implementación del sistema. Se refleja la forma en la cual se establecerá la conexión a los repositorios de patrones de diseño utilizando el protocolo REST. Además, se evidencian los requisitos funcionales y no funcionales identificados en la investigación, así como las Historias de Usuario (HU) que describe cada requisito funcional.

Capítulo 3: Implementación y validación del sistema de recomendación

Se describe la implementación de las principales funcionalidades del sistema y los estándares de codificación utilizados. De las pruebas existentes para verificar si una aplicación cumple con los requerimientos identificados por una investigación, las empleadas en el presente trabajo fueron: la de caja blanca y la de caja negra. También se utiliza la técnica de ladov, la cual se emplea para determinar el nivel de satisfacción que tienen los usuarios con respecto al sistema de recomendación propuesto.

Capítulo 1. Marco teórico sobre los sistemas de recomendación y los patrones de diseño de recursos educativos

En el presente capítulo se describen conceptos que permiten la comprensión de las principales características de los recursos educativos, los patrones de diseño, los lenguajes, los patrones pertenecientes a catálogos y los sistemas de recomendación. Además, se reseñan las herramientas y tecnologías utilizadas para el desarrollo del sistema.

1.1 Recursos educativos

Los recursos educativos son materiales con intención didáctica para apoyar el desarrollo de los procesos de enseñanza - aprendizaje. Se entiende como cualquier instrumento u objeto que pueda servir como recurso que, mediante la manipulación, observación o lectura, ofrezcan oportunidades de aprender sobre un tema en específico [11].

Los recursos educativos incluyen: cursos completos, materiales para cursos, módulos, libros de texto, vídeos, pruebas, software y cualquier otra herramienta que contribuya el apoyo para acceder al conocimiento [12] [13].

Los recursos educativos pueden estar compuestos por [2] [3] [11] [12]:

- **Contenidos educativos:** programas educativos, módulos de contenido, objetos de aprendizaje, materiales multimedia (texto, sonido, video, imágenes, animaciones), exámenes, publicaciones periódicas (diarios y revistas), etc.
- **Herramientas:** software para apoyar la creación, uso y mejoramiento de contenidos educativos. Esto incluye herramientas y sistemas para: crear, registrar y organizar contenido; gestionar el aprendizaje y desarrollar comunidades de aprendizaje en línea.
- **Recursos de implementación:** licencias de propiedad intelectual que promuevan la publicación abierta de materiales, principios de diseño, adaptación, localización de contenido. Por lo general, quienes crean recursos educativos, permiten que cualquier persona use sus materiales, los modifique, los traduzca o los mejore y, además, que los comparta con otros. Se deben tener en cuenta que algunas licencias restringen las modificaciones (obras derivadas) o el uso comercial.

- **Enlaces externos:** observatorios y centros de información para la promoción del uso, creación y difusión de estos recursos.

Los recursos educativos cumplen las siguientes características:

- Su objetivo es apoyar el proceso de enseñanza - aprendizaje a partir de su propósito educativo.
- Según la licencia bajo la cual se encuentren, se pueden reutilizar como un todo o solamente una de sus partes.
- Son volátiles pues permiten la adaptación, mezcla y mejora.

1.2 Patrones de diseño

En 1977 se utilizó por primera vez el término de patrones de diseño afirmado por Christopher Alexander: "cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma" [7].

Para facilitar y volver más eficiente la creación de un recurso educativo, se ha planteado el uso de plantillas independientes del contenido temático o patrones de diseño de recursos educativos. Estas plantillas establecen esquemas de entrelazado de información, actividades y evaluación a partir de las cuales se puede plantear un espectro amplio de recursos educativos con idéntica lógica interna que habiliten una diversidad de contenidos[8].

Los patrones de diseño se pueden visualizar como una solución a un problema en un contexto dado, tres de sus aspectos fundamentales que se destacan son los siguientes [9]:

- **Nombre del patrón:** Debe ser un nombre significativo y descriptivo en cuanto al problema tratado, a ser posible, formado por una sola palabra o expresión descriptiva que resuma su contenido.
- **Problema:** Descripción del problema cuya solución se propone. Suele incluir una descripción del propósito del problema que se presenta y si la solución propuesta es aplicable en cuanto a su contexto.

- **Solución:** Descripción clara de cómo alcanzar el resultado deseado. Es el equivalente a las instrucciones de aplicación del patrón. Es conveniente recoger uno o más ejemplos claros de aplicación del patrón (su contexto inicial de aplicación, el modo de aplicación, los resultados). Recoger un ejemplo claro ayuda a entender el uso y aplicación del patrón.

La utilización de patrones de diseño proporciona numerosos beneficios para confección de un recurso educativo:

- Brindan una fácil reutilización de diseños para crear recursos educativos.
- Los diseñadores pueden acceder de forma sencilla a técnicas o diseños previos exitosos.
- Permiten seleccionar diferentes alternativas de solución para un problema en específico.
- Mejora notablemente la documentación y mantenimiento de sistemas existentes.

Actualmente, existen numerosos repositorios donde se almacenan patrones de diseño, por ejemplo, Pedagogical Pattern (PPP), E-LEN, PCeL y REPLIKA. Algunos de ellos agrupan patrones en colecciones (lenguajes y catálogos).

1.3 Lenguajes de patrones de diseño

En el libro “*A Pattern Language*” de Christopher Alexander está implícita la idea de que los patrones deben estar organizados por el tipo de problema que resuelven. Los patrones de diseño deben estar relacionados los unos con los otros para poder formar un lenguaje de patrones, por ello, cada patrón debe indicar su relación con otros patrones [7].

Existen muchas colecciones de patrones de diseño, a menudo los autores intentan tejerlos en un lenguaje. Ellos los tratan como conjuntos de patrones interrelacionados que se complementan entre sí que respalda un proceso de desarrollo generativo y específico de dominio [6].

Así como un patrón de diseño independiente es mucho más que “una solución a un problema que se presenta en un contexto específico”, un lenguaje de patrones es mucho más que una

red de patrones de diseño estrechamente enlazados que definen un proceso para resolver sistemáticamente un conjunto de problemas comunes e interdependientes [14].

Los patrones de diseño que forman parte de un lenguaje de patrones se almacenan en un repositorio de patrones accesible a través de la web. El repositorio proporciona un índice de todos los patrones de diseño y contiene vínculos de hipermedias que permiten al usuario entender las colaboraciones entre esos patrones [15].

Por esta razón, se puede concluir que los lenguajes de patrones de diseño son un conjunto de patrones que por sí solos resuelven problemas de diseño y al relacionarse resuelven un problema aún mayor.

1.4 Catálogos de patrones de diseño

Un catálogo de patrones permite agrupar a los patrones teniendo en cuenta cualidades en común que tengan éstos. Suelen relacionarse generalmente por el tipo de problema que resuelven, y pueden proporcionar, de conjunto, una solución [16].

Los catálogos son un grupo de patrones clasificados por uno o más criterios y relacionados entre sí, los cuales pueden ser utilizados de forma conjunta o independiente. Estos criterios permiten organizar los patrones de diseño en grupos que comparten el mismo conjunto de propiedades, y dependiendo de los criterios elegidos se pueden definir esquemas de clasificación con diferentes dimensiones. Los esquemas de clasificación ponen de manifiesto las principales cualidades de los patrones y ayudan a reducir el tamaño del espacio de búsqueda [17].

La definición o selección de los criterios de organización para un determinado catálogo dependerá del objetivo de una clasificación específica, en este caso sería, ayudar a los diseñadores de recursos educativos seleccionar los patrones de diseño pertenecientes a ese catálogo que responden ante un problema de diseño al que se están enfrentando. Por esta razón, los catálogos son un conjunto de patrones que se unen respondiendo a una característica común, generalmente el tipo de problema que resuelven.

1.5 Sistemas de recomendación

Los sistemas de recomendación surgieron a mediados de la década de los 90 con el fin de brindarles a los usuarios aquellos resultados de búsquedas de información cercanos a sus necesidades.

Un sistema de recomendación puede realizar predicciones a partir del hecho que a un usuario le guste o no cierto contenido al que podría acceder. Tienen la capacidad de identificar preferencias y sugerir ítems relevantes para cada usuario; por ello se necesita de perfiles que almacenen la información y las preferencias de cada usuario [18].

Estos sistemas surgen como respuesta a la necesidad de contar con una herramienta de personalización de contenidos, utilizando información histórica del usuario para recomendarle elementos que le agraden [19].

Además, los sistemas de recomendación intentan determinar qué productos o servicios son los adecuados que se pueden basar en las preferencias y limitaciones del usuario. Con el fin de completar tareas, recogen del usuario sus preferencias, que se expresan explícitamente o se deducen interpretando las acciones del usuario [20].

Los sistemas de recomendación tienen varias clasificaciones entre las que se encuentran [21]:

- Sistemas de recomendación colaborativa.
- Sistemas de recomendación basados en contenido.
- Sistemas de recomendación basados en conocimiento.
- Sistemas de recomendación híbridos.

La finalidad de un sistema de recomendación es predecir la valoración que un usuario va a hacer de un ítem que todavía no ha evaluado. Esta valoración se genera al analizar una de dos cosas, o las características de cada ítem, o las valoraciones de cada usuario a cada ítem, y se usa para recomendar contenido personalizado a los usuarios.

Sistemas de recomendación colaborativa

El filtrado colaborativo es una técnica empleada por los sistemas de recomendación que utiliza la información de preferencias y calificación de un grupo de usuarios respecto a contenidos almacenados en diversas fuentes, con el fin de predecir o inferir la preferencia de un usuario en particular sobre un ítem y a partir de esto generar una recomendación acertada [22].

Los sistemas de recomendación colaborativa agregan clasificaciones o recomendaciones de objetos, donde reconocen las características comunes entre los usuarios de la base de evaluaciones, y generan nuevas recomendaciones basadas en comparaciones entre esos usuarios [23].

Además, estos sistemas funcionan recopilando comentarios de los usuarios en forma de calificaciones de elementos en un dominio dado y explotando similitudes en el comportamiento de calificación entre varios usuarios para determinar cómo recomendar un elemento en específico [19].

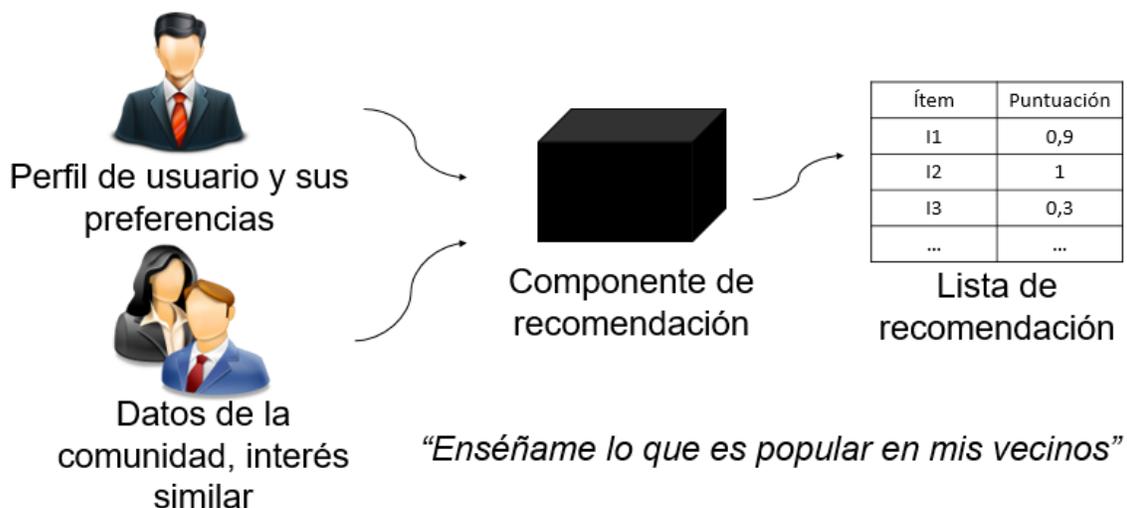


ILUSTRACIÓN 1. SISTEMAS DE RECOMENDACIÓN COLABORATIVA

Sistemas de recomendación basados en contenido

La recomendación basada en contenido es una consecuencia y continuación de la investigación de filtrado de información. En estos sistemas, los objetos de interés se definen por sus características asociadas. (...) Los sistemas de este tipo aprenden de un perfil según los intereses del usuario basado en las características presentes en los objetos que el usuario

ha calificado, llaman a esto “correlación de elemento a elemento”. (...) Al igual que en el caso de colaboración, los perfiles de usuario son modelos a largo plazo y se actualizan a medida que se observan más pruebas sobre las preferencias del usuario [21].

Michael J. Pazzani y Daniel Billsus de la Universidad Rutgers plantean que: “los sistemas de recomendación basados en contenido analizan las descripciones de los ítems para identificar los que son de particular interés para el usuario”. Por lo que los detalles del sistema de recomendación definen términos según la representación de esos ítems [24].

Además, un sistema de recomendación basado en contenido permite sugerir al usuario nuevos productos en función de su similitud con el contenido (descripción) de otros objetos que éste ha juzgado anteriormente [25].

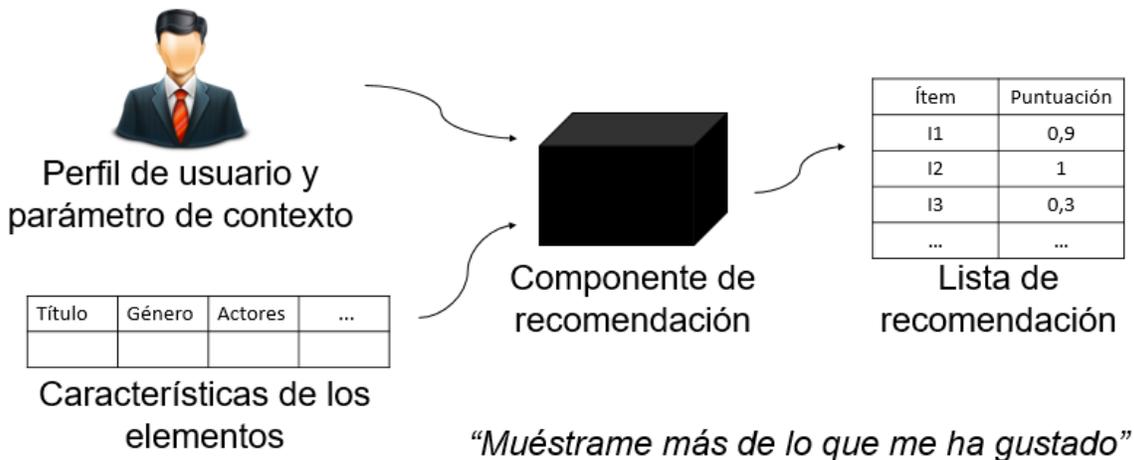


ILUSTRACIÓN 2. SISTEMAS DE RECOMENDACIÓN BASADO EN CONTENIDO

Sistemas de recomendación basados en conocimiento

La recomendación basada en conocimiento intenta sugerir objetos basados en inferencias sobre las necesidades y preferencias de un usuario. Los enfoques basados en conocimiento se distinguen porque tienen conocimiento funcional: sobre cómo un artículo en particular satisface una necesidad particular del usuario, por lo tanto, pueden razonar sobre la relación entre una necesidad y una posible recomendación. El perfil de usuario puede ser cualquier estructura de conocimiento que respalde esta inferencia [21].

Los sistemas de recomendación basados en conocimiento utilizan técnica de razonamiento basado en casos (...). En estos sistemas los usuarios dan un ejemplo del tipo de producto que están buscando, y el sistema buscará y recomendará productos similares al ejemplo dado. Como es de suponer, la mayoría de las veces el usuario lo que busca no es un producto exactamente igual al producto ejemplo, si no un producto parecido a este. Por tanto, estos sistemas deben permitir que los usuarios refinen sus búsquedas declarando o modificando algunas de los atributos del ejemplo dado [26].

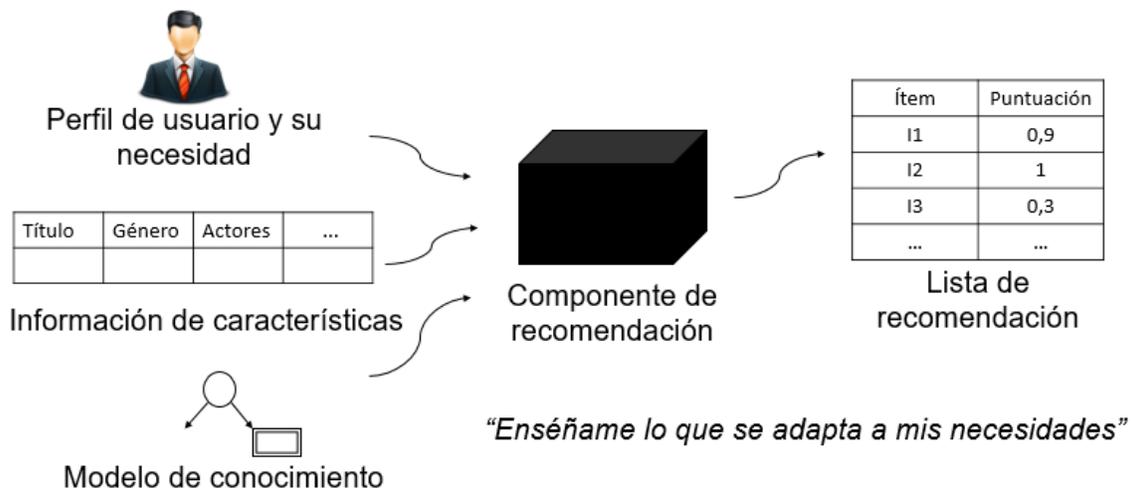


ILUSTRACIÓN 3. SISTEMAS DE RECOMENDACIÓN BASADO EN CONOCIMIENTO

Sistemas de recomendación híbrida

Los sistemas de recomendación híbrida combinan dos o más técnicas de recomendación para obtener un mejor rendimiento con menos de los inconvenientes de cualquier individuo. Comúnmente, el filtrado colaborativo se combina con alguna otra técnica con el fin de recomendar un ítem que tenga un alto nivel de aceptación de otros usuarios [23].

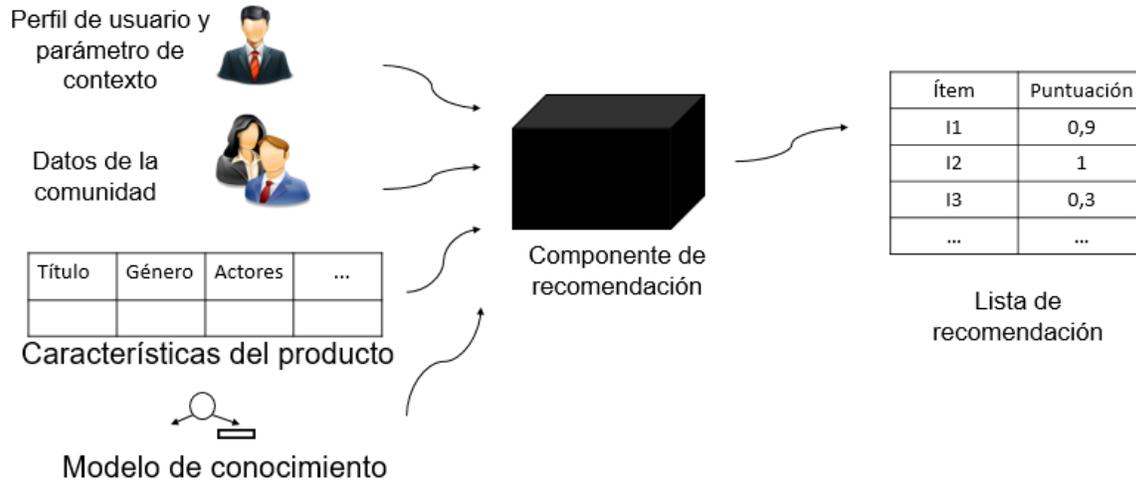


ILUSTRACIÓN 4. SISTEMAS DE RECOMENDACIÓN HÍBRIDA

Fundamentación de la selección

En la presente investigación se aplicará la recomendación híbrida, entre colaborativa y basado en conocimiento.

Entre las características más relevantes por las que está compuesto la recomendación basado en conocimiento se utilizan: generar 0, 1 o múltiples soluciones, representar el conocimiento, el cuestionamiento, la emisión de conclusiones, y el razonamiento con base en probabilidades, que recomendará patrones de diseño independientes, lenguajes o patrones pertenecientes a un catálogo que más posibilidades tengan de resolver el problema de diseño de recursos educativos. También proporciona la búsqueda de semejanzas de los textos que se desean comparar a partir de métricas de similitud entre cadenas.

Para realizar la comparación del problema entrado al sistema con la descripción del problema almacenado en los repositorios de patrones de diseño se aplicó la técnica de n-gramas contextuales, que permitió comparar las cadenas de textos y recomendar el patrón que más se asemeja al problema que se desea resolver.

Para realizar una recomendación colaborativa se tiene en cuenta el nivel de satisfacción de los usuarios al interactuar con determinados patrones donde se analizan varios parámetros (cantidad de descargas, cantidad de recomendaciones y promedio de evaluaciones).

1.6 Técnica de n-gramas contextuales

Al realizar la comparación entre textos (el introducido por el usuario y la descripción del problema de diseño de recursos educativos almacenado del patrón) se hace necesario el estudio de cómo poder efectuar esta comparación.

Por tal motivo, se investigó acerca de los n-gramas contextuales, que tienen como principal característica representar la esencia del contexto con un reducido número de caracteres. Estos n-gramas se obtienen a partir de un texto, donde se agrupan en una cadena de n palabras (con un alto significado en cuanto al contexto). Para adquirir estas cadenas, se realizan un conjunto de pasos entre los que se encuentran la eliminación de palabras vacías y caracteres aislados, extracción del lexema y ordenación de cada n-grama [27], [28], [29].

Varios estudios realizados por un conjunto de autores [27], [28], [29] se aproximan al buen uso de n-gramas contextuales, ellos arriban que según el grado de conveniencia de n-gramas, estos pudieran mostrar mejores resultados al realizar ciertas comparaciones. Entre los n-gramas más usados se encuentran los bigramas ($n=2$) y los trigramas ($n=3$ con una mejor aceptación). Pero también concuerdan que la utilización de trigramas brinda una amplia cobertura, precisión y granularidad con respecto al empleo de bigramas.

Para conseguir que el trigramas utilizado en la presente investigación contenga la mejor definición de la esencia del contexto y sea especialmente útil para comparar dos textos, se llevan a cabo cinco pasos [10], [27]:

1. Conversión a minúsculas es una práctica común.
2. Eliminación de las palabras vacías conocidas también como stopwords.
3. Eliminación de palabras de un solo carácter. Este paso permite que no se afecte la comparación entre los textos ante el posible cambio de orden en la cadena de n-gramas por ejemplo cuando se hace uso de conjunciones “y” y “o”.
4. Reducción a la raíz de las palabras, de modo que el género o tiempo verbal utilizado no afecte la búsqueda en la base de casos. Este paso contribuye a que no se afecte la comparación de textos en los casos de sustitución de palabras por sus derivadas.

5. Combinación interna de los tokens del n-grama, procesándose como representante canónico del conjunto de sus posibles permutaciones. Este paso anula el efecto de un posible cambio de orden de las palabras al comparar dos frases.

1.7 Búsquedas federadas

Los datos para realizar la recomendación de patrones de diseño de recursos educativos son: los parámetros para determinar el nivel de satisfacción por parte de los usuarios al interactuar con determinados patrones y el problema que resuelve el patrón de diseño analizado en determinado momento. Para la obtención de estos datos se hará uso de la Interfaz para Programar Aplicaciones (API, por sus siglas en inglés). Las búsquedas en los diferentes repositorios de patrones de diseño se pueden llevar a cabo de una manera paralela, realizando búsquedas federadas.

Una búsqueda federada permite buscar objetos simultáneamente en múltiples fuentes. Dependiendo de cómo esté configurada la búsqueda, las fuentes de esta pueden incluir entidades tales como base de datos, sitios web, repositorios, entre otros [30].

Con la federación, la consulta se puede realizar en el índice de contenido local o se puede reenviar a un repositorio de contenido externo, donde el motor de búsqueda procesa ese contenido en el repositorio. A continuación, el motor de búsqueda del repositorio devuelve los resultados al servidor o aplicación que realizó la petición de búsqueda.[30]

La relevancia de los resultados de los productos que se acceden mediante búsquedas federadas depende de las búsquedas simultáneas realizadas a las diferentes fuentes externas [31].

1.8 Análisis de soluciones similares

A continuación, se analizan algunos sistemas de recomendación para conocer su funcionalidad y sus propuestas de solución.

Se analizó el trabajo titulado “Recomendación de objetos de aprendizaje basado en el perfil del usuario y la información de atención contextualizada” conciliado en la Universidad Oberta de Catalunya que propone un prototipo de sistema de recomendación de objetos de aprendizaje que ha sido utilizado para experimentar con tres diferentes algoritmos de

recomendación basados en el perfil del usuario con el fin de determinar y comparar la precisión de sus resultados [22].

El “Modelo multi-agente para recomendación híbrida de objetos de aprendizaje” desarrollado en la Universidad Nacional de Colombia, propone a través de un enfoque multi-agente una recomendación de objetos de aprendizaje basados en un perfil. Realizando una búsqueda en repositorios locales y remotos, o en federaciones de repositorios, accesibles vía web y con metadatos descriptivos de estos objetos. La recomendación la realiza un solo agente que hace un filtrado. En este trabajo se extiende el agente de recomendación a un sistema de cinco agentes y que se puede extender a otros más, donde cada uno de los agentes es responsable de una técnica de recomendación y hay un agente deliberativo que decide que recomendación es más adecuada para el usuario [32].

Otro sistema analizado es el “Sistema de recomendación basado en k-nn para condiciones de incertidumbre en un Sistema Tutor Inteligente” desarrollado en la universidad que propone una técnica para el desarrollo de un módulo del estudiante con el uso de redes bayesianas y filtrado colaborativo, vinculando las fases a través de una medida de efectividad generada durante la evaluación, llamada coeficiente de efectividad [33].

Por último, se analizó el “Sistema de Recomendación de patrones de diseño para Recursos Educativos Abiertos” desarrollado en la universidad que contribuye a la selección de patrones adecuados para dar solución a un problema de diseño de un recurso educativo abierto. Utilizando el razonamiento basado en casos para el desarrollo de la aplicación, siendo la base de casos el repositorio de patrones. Para realizar la comparación del problema entrado al sistema con el problema almacenado en la base de casos se aplicó la técnica de n-grama contextual, que permitió comparar las cadenas de textos y recomendar el patrón que más se asemeja al problema que se desea resolver.

TABLA 1. CUADRO COMPARATIVO ENTRE LOS SISTEMAS SIMILARES

Trabajo	Tratamiento de agrupaciones	Tipo de recomendación				Contenido
		Conocimiento	Contenido	Colaborativa	Híbrido	
Recomendación de objetos de aprendizaje basado en el perfil del usuario y la información de atención contextualizada	No	No	Sí	Sí	Sí	Atención contextualizada
Modelo multi-agente para recomendación híbrida de objetos de aprendizaje	No	Sí	Sí	Sí	Sí	Sistema multi-agente
Sistema de Recomendación basado en k-nn para condiciones de incertidumbre en un Sistema Tutor Inteligente	No	No	No	Sí	No	Redes bayesianas
Sistema de Recomendación de patrones de diseño para Recursos Educativos Abiertos	No	Sí	No	No	No	Patrones de diseño

El estudio de estas soluciones evidenció que solo responden a las necesidades particulares de cada institución para las cuales fueron creadas. A partir del cuadro comparativo se puede

apreciar que la mayoría de los sistemas estudiados utilizan uno o varios tipos de recomendación, de los cuales, el que predomina es la recomendación colaborativa una de las escogidas para emplearla en la presente investigación. Además, estos sistemas no tienen presente algún tipo de agrupamiento con respecto al contenido con los que trata cada uno. Por lo antes descrito se hace necesario la implementación de un nuevo sistema de recomendación de patrones de diseño de recursos educativos que incluya el tratamiento de agrupaciones y la calidad percibida por los usuarios.

1.9 Herramientas y tecnologías a utilizar

Para darle cumplimiento al objetivo planteado en el presente trabajo investigativo es necesario explicar de forma detallada la metodología, herramientas y tecnologías a utilizar.

Metodología de desarrollo de software

Una metodología es un conjunto de filosofías, etapas, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información [34].

Se determinó guiar el proceso de desarrollo del sistema a través de la metodología Proceso Unificado Ágil (AUP) en su variante UCI. Esta variante realizada por la UCI a la metodología ágil AUP está definida como el documento rector de la actividad productiva en la universidad, además es utilizada por el grupo de investigación al cual pertenece la presente investigación, de forma tal que se adapte al ciclo de vida definido para dicha actividad. La misma está compuesta por tres fases: Inicio, Ejecución y Cierre que en su conjunto ayudan a mejorar el proceso de desarrollo de software.

Esta metodología contiene las características de las cuatro fases (Inicio, Elaboración, Construcción y Transición) propuestas en AUP. De estas fases se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes tres fases de AUP en una sola, en la fase de Ejecución y se agrega la fase de Cierre [35].

A continuación, se describen los objetivos de las fases definidas [35]:

- **Inicio:** Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
- **Ejecución:** En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, se obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto. Durante esta fase el software es transferido al ambiente de los usuarios finales o entregado al cliente junto con la documentación. Además, en esta transición se capacita a los usuarios finales sobre la utilización de la aplicación.
- **Cierre:** En el cierre se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Para modelar el sistema se utilizó el cuarto escenario de esta metodología el cual define que su aplicación se debe realizar en proyectos que hayan evaluado el negocio a informatizar y como resultado se obtenga un negocio muy bien definido donde el cliente estará siempre acompañando al equipo de desarrollo. Este escenario propone elaborar HU en las cuales se debe establecer conversaciones acerca de las necesidades de los clientes. Además, se recomienda en proyectos no muy extensos, pues una HU no debe poseer demasiada información.

Lenguaje de programación: Java

Java, desarrollado por Sun Microsystems en 1995, es un completo lenguaje de programación orientado a objetos diseñado para distribuir contenidos a través de una red. Una de sus principales características es que permite operar de forma independiente de la plataforma y del sistema operativo que se esté utilizando. Esto quiere decir que permite crear una aplicación que podrá descargarse de la red y funcionar posteriormente en cualquier tipo de plataforma de hardware o software [36].

Los programas desarrollados en Java se compilan a un lenguaje intermedio, denominado Bytecode. Este código es interpretado por la máquina virtual de Java del entorno de ejecución (JRE, según sus siglas en inglés) y así se consigue la portabilidad en distintas plataformas. El JRE es un intermediario entre el código Bytecode y los distintos sistemas operativos existentes en el mercado. Un programa Java compilado en Bytecode se puede ejecutar en sistemas operativos como Windows, Linux, iOS o Android utilizando el JRE apropiado. Una de las características más importantes de los lenguajes de programación modernos es la portabilidad. Un programa es portable cuando es independiente de la plataforma y puede ejecutarse en los principales sistemas operativo y dispositivo físico. [37] [38].

El lenguaje de programación que se utilizará es el Java, pues este lenguaje se predefinió a utilizar en esta investigación porque es el lenguaje con que se desarrolló el sistema REPREA.

Visual Paradigm

Visual Paradigm es una herramienta CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora) que soporta el ciclo de vida completo en el desarrollo de software: análisis y desarrollos orientados a objetos, construcción, prueba y despliegue [39] [40].

Entre las características fundamentales que determinaron su selección se destaca que: brinda apoyo adicional en cuanto a el flujo de información del sistema, así como su comportamiento, tiene disponibilidad en múltiples plataformas, brinda la posibilidad de intercambiar información mediante la importación y exportación de ficheros; y permite crear los prototipos de usuario con los que se guiará el proceso de desarrollo del sistema.

Gestor de base de datos: PostgreSQL

Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos, contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente [41].

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado [42].

Algunas características de PostgreSQL [42]:

- Integridad referencial.
- Replicación asincrónica/sincrónica.
- Regionalización por columna.
- Múltiples métodos de autenticación.
- Acceso encriptado.
- Amplia documentación.
- Disponible para Linux y UNIX y Windows 32/64bit.

Algunas de las ventajas de PostgreSQL [43]:

- Fácil de Administrar.
- Su sintaxis SQL es estándar y fácil de aprender.
- Multiplataforma.
- Capacidades de replicación de datos.
- Soporte empresarial disponible.

Algunas de las desventajas de PostgreSQL [43]:

- En comparación con MySQL es más lento en inserciones y actualizaciones, ya que cuenta con cabeceras de intersección que no tiene MySQL.
- Consume más recursos que MySQL.
- La sintaxis de algunos de sus comandos o sentencias no es nada intuitiva.

Se predefinió el gestor de base de datos PostgreSQL, también fue utilizado por los desarrolladores del sistema antes mencionado.

1.10 Interoperabilidad

La interoperabilidad entre aplicaciones informáticas es cada vez más necesaria en las organizaciones, la competitividad induce a prestar nuevos y mejores servicios los cuales son generados a través de los sistemas de información existentes en las organizaciones. Las diferentes fuentes de información debido a varias aplicaciones de diversos proveedores, en ocasiones provoca dificultad para unir información y agilizar las operaciones dentro de una organización. Dicho problema se puede solventar con la adquisición de una API a través del área de sistemas de la organización [44] [45].

La interoperabilidad es la capacidad de dos o más sistemas con iguales o diferentes características para intercambiar entre sí información y poder usarla. Esta es una característica muy importante pues permite la comunicación entre sistemas independientemente de las propiedades y las plataformas sobre las que se instalen. La interoperabilidad se consigue mediante el uso de estándares y protocolos abiertos como es el caso de HTTP, XML, WSDL, etc. [44] [45].

1.11 Conclusiones parciales

El análisis de las características de los recursos educativos y de los patrones de diseño permitió comprender de manera más fácil el problema a resolver de la presente investigación. Además, con el estudio minucioso de los sistemas de recomendación, sus principales elementos y su clasificación, se determinó utilizar la recomendación híbrida, entre las recomendaciones colaborativa y basadas en conocimiento. También se utiliza como lenguaje de programación el lenguaje Java, la herramienta de modelado el Visual Paradigm y como gestor de base de datos PostgreSQL.

Capítulo 2. Sistema de recomendación de patrones de diseño de recursos educativos

En el presente capítulo se refleja el cumplimiento del objetivo trazado en la investigación. Se describe la arquitectura del sistema de recomendación, así como los patrones arquitectónicos y de diseño que son utilizados en la implementación de la aplicación.

2.1 Propuesta de solución

Para dar solución a la problemática planteada en la presente investigación se implementa el Sistema para recomendar patrones de diseño de recursos educativos basado en el tratamiento de agrupaciones y la calidad percibida por los usuarios, un sistema que le permite a los usuarios seleccionar patrones de diseño pertenecientes a catálogos, lenguajes de patrones de diseño o patrones de diseño independientes que le brinde una solución al problema de diseño que se desea resolver de recursos educativos.

Cuando se introduce el problema de diseño de recursos educativos, el sistema analiza el agrupamiento de los patrones, teniendo en cuenta que la solución pudiera ser a través de un patrón perteneciente a un catálogo, un lenguaje o un patrón independiente. Para ello, realiza la comparación entre los textos de las descripciones de los problemas, el introducido por el usuario y el correspondiente a los patrones almacenados. Una vez comparadas las descripciones de los problemas y que la medida de semejanza sea igual o superior al umbral definido se calculan varios parámetros que permiten medir la calidad percibida por los usuarios, los cuales son: la descarga, la recomendación y la evaluación.

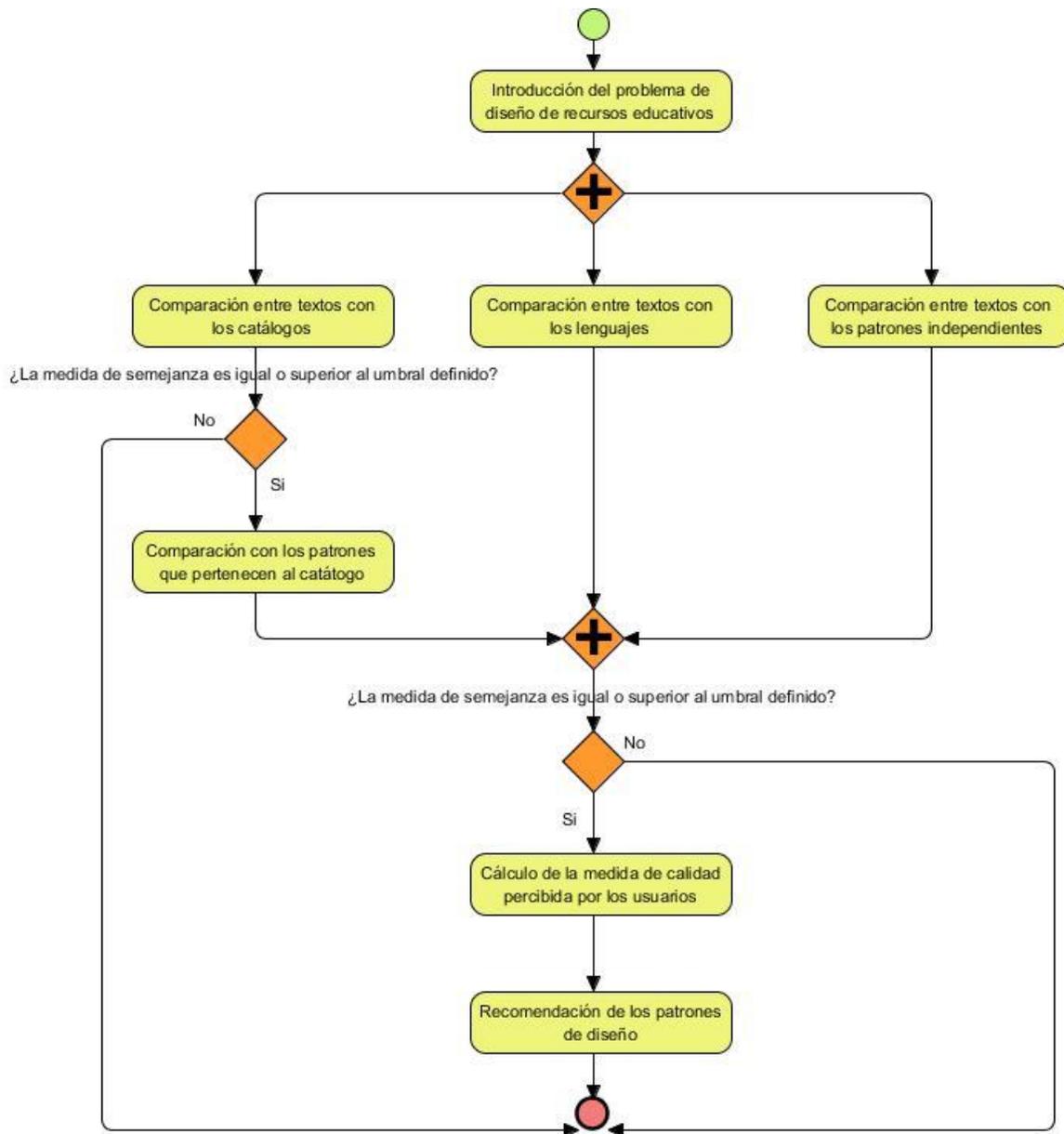


ILUSTRACIÓN 5. DIAGRAMA DE PROCESOS DEL FLUJO DE DATOS DEL SISTEMA

A continuación, se muestra el pseudocódigo del principal proceso que realiza el sistema:

Función Recomendación (cadena problema): Colección

Inicio

Obtener datos de repositorio

Obtener patrones independientes

Obtener patrones pertenecientes a catálogos

Obtener lenguajes

Inicializar colección vacía

Adicionar patrones independientes a la colección.

Adicionar patrones pertenecientes a catálogos a la colección

Adicionar lenguajes a la colección

Colección semejanzas \leftarrow semejanza_entre_textos (colección)

Inicializar Colección de recomendados

Inicializar contador en 0

Mientras contador < longitud de la colección

Inicio

Elemento recibe colección (contador)

Calcular calidad percibida para el elemento

Si calidad percibida ≥ 0.8

Inicio

Adicionar elemento a Colección de recomendados

Fin Si

Fin Mientras

Ordenar Colección de recomendados

Retornar Colección de recomendados

Fin

2.1.1 Comparación entre textos

Para realizar la comparación entre los textos de las descripciones de los problemas, el introducido por el usuario y el correspondiente al patrón, se identificaron algunas características del problema. Para ello en la presente investigación se utiliza el mismo procedimiento que desarrollaron los autores del sistema REPREA, la recuperación de información basada en la técnica de n-gramas contextuales ya documentadas en el epígrafe 1.6.

Para determinar si el problema entrado s puede ser semejante al problema almacenado d se plantea la siguiente ecuación: la medida de semejanza S es útil cuando los conjuntos de n-gramas a comparar provienen de textos de longitud equiparable. Considerando un problema s y uno almacenado d , la semejanza se define por medio de la ecuación:

$$S(s | d) = \frac{|N(d) \cap N(s)|}{|N(d) \cup N(s)|}$$

ILUSTRACIÓN 6. ECUACIÓN DE MEDIDA DE SEMEJANZA

Donde $N(d)$ es el conjunto de n-gramas en la cadena de texto d del problema almacenado y $N(s)$ es el conjunto de n-gramas en la cadena de texto s que será entrada al sistema.

La medida de semejanza toma valores entre el intervalo $[0,1]$, por lo que se define un umbral dentro de este intervalo tal que, al ser superado, se considere que el texto entrado por parámetro es semejante al texto almacenado. El umbral definido para la investigación es de 0,8 [10].

2.1.2 Calidad percibida por los usuarios

Para realizar la recomendación de patrones de diseño de recursos educativos es necesario gestionar información sobre los diferentes elementos básicos que maneja el sistema: las preferencias de los usuarios, los catálogos, los lenguajes, los patrones de diseño y los repositorios de patrones.

Los elementos a tener en cuenta para calcular la calidad percibida por los usuarios son:

- **Cantidad de descargas:** con este parámetro se obtendrá la cantidad de veces que los usuarios han descargado determinado patrón, lenguaje o patrones pertenecientes a catálogos.
- **Cantidad de recomendaciones:** con este parámetro se obtendrá la cantidad de veces que los usuarios han recomendado con anterioridad determinado patrón, lenguaje o patrones pertenecientes a catálogos.
- **Promedio de evaluaciones:** con este parámetro se obtendrá el promedio de evaluaciones acerca del criterio que tengan los usuarios de determinado patrón, lenguaje o patrones pertenecientes a catálogos.

Para obtener la información de los parámetros se realiza a través de los resultados de interacción de los usuarios con determinado patrón de diseño mediante un indicador de interacción. El cálculo de este indicador fue guiado por los aportes dados en [46], [47] y se calcula para aquellos lenguajes, patrones pertenecientes a un catálogo y patrones independientes seleccionados en la fase anterior y toma valores entre 0 y 1, se calcula de la siguiente forma:

$$INT(p) = \sum_{i=1}^v \frac{0,5(\frac{E_i}{5}) + 0,3R_i + 0,2D_i}{v}$$

ILUSTRACIÓN 7. ECUACIÓN PARA DETERMINAR LA CALIDAD PERCIBIDA POR LOS USUARIOS

Donde:

- v es la cantidad de veces que el patrón ha sido visualizado.
- E_i es la evaluación otorgada al patrón por el usuario en la visualización i , toma valor entero entre 1 y 5.
- R_i y D_i toman valor 1 si el usuario ha recomendado y descargado, respectivamente, el patrón en la visualización i y valor 0 en caso contrario.

Una vez concluida la comparación entre los textos, el introducido por el usuario y la descripción correspondiente al patrón almacenado, se realiza el cálculo del índice de interacción de los usuarios

con determinado patrón, para establecer un orden de prioridad para mostrarle finalmente a los usuarios cuáles son los lenguajes, patrones pertenecientes a un catálogo y patrones independientes que le dan solución al problema de diseño introducido por ellos.

2.2 Arquitectura

La arquitectura de software se ha consolidado como una disciplina que intenta contrarrestar los efectos negativos que pueden surgir durante el desarrollo de un sistema, ocupando un rol significativo en la estrategia de negocio de una organización que basa sus operaciones en el software [48].

Otra fuente que define a la arquitectura de software es la IEEE 1471-2000 caracterizándola como una organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución.

Al diseñar una arquitectura de software se crean y representan componentes que interactúan entre sí, con responsabilidades específicas y se organizan de forma tal que se logren los requerimientos establecidos. Se puede partir con patrones de soluciones probados que se conocen con el nombre de estilos arquitectónicos, patrones arquitectónicos y patrones de diseño [48].

Entre las arquitecturas más destacadas se puede mencionar: la arquitectura cliente-servidor, la arquitectura orientada a servicios (SOA, por las siglas en inglés de service-oriented architectures), la arquitectura orientada a objetos, la arquitectura Modelo-Vista-Controlador (MVC), la arquitectura basada en componentes y la arquitectura en capas.

Pressman define algunas de las arquitecturas antes mencionadas como por ejemplo [15]:

- La arquitectura orientada a objetos: se basa en los componentes de un sistema que incluyen datos y las operaciones que deben aplicarse para manipularlos. La comunicación y coordinación entre los componentes se consigue mediante la transmisión de mensajes.
- La arquitectura en capas: se basa en la estructura básica de capas. Se define un número de capas diferentes, cada una ejecuta operaciones que se aproximan progresivamente al conjunto de instrucciones de máquina. En la capa externa, los

componentes atienden las operaciones de la interfaz de usuario. En la interna, los componentes realizan la interfaz con el sistema operativo. Las capas intermedias proveen de utilerías y funciones de software de aplicación.

El MVC es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos [49]:

- **Modelo:** contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- **Vista:** compone la información que se envía al cliente y los mecanismos interacción con éste.
- **Controlador:** actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

Según Sommerville define a SOA como una forma de desarrollar sistemas distribuidos en la que los componentes del sistema son servicios independientes y se ejecutan en computadoras distribuidas geográficamente [50]. Los protocolos estándar basados en XML, tales como SOAP y WSDL, se diseñaron para dar soporte al servicio de comunicación e intercambio de información [51].

En la presente investigación se utiliza la arquitectura MVC. Se emplea para tener una mejor estructura del sistema, así separando el código de las vistas. Para ello se emplea el patrón arquitectónico con el mismo nombre MVC.

2.3 Conexión con repositorios

Para establecer la conexión a los repositorios de patrones de diseño se utiliza la transferencia de estado representacional (REST en sus siglas en inglés).

REST utiliza básicamente la arquitectura de la Web para compartir datos y recursos, aunque REST no es una tecnología ni un protocolo es una solución efectiva, además es totalmente independiente de cualquier tecnología y plataforma, que combina las ventajas de la comunicación mediante XML con la simplicidad de las transacciones a través del protocolo

HTTP. Este protocolo define métodos de comunicación y procesamiento de transacciones mediante los métodos (GET, POST, PUT, DELETE). Para REST todo es un recurso, que puede identificarse y accederse por medio de su URI (Identificador de recursos uniforme) por tanto estos pueden manipularse por medio de las operaciones anteriormente mencionadas [52] [53] [54].

Para la utilización de REST no es obligatorio seguir todos sus principios, esto ha dado pie a una definición que acoge los sistemas basados en estos principios de manera estricta conocida como RESTful, pues es un servicio con una mejor alternativa para integrar sistemas [52] [53] [54].

Según el *World Wide Consortium* (W3C), uno de los organismos responsables de su arquitectura y que regulan la reglamentación de los mismos definen a un servicio web como un sistema de software diseñado para mantener comunicaciones interoperables máquina a máquina sobre una red [55].

Además, los servicios web son API's a las que se puede acceder dentro de una red y se ejecutan en el sistema que las contiene. Java posee una API de desarrollo para servicios RESTful llamada JAX-RS que define algunas anotaciones para el desarrollo de estos servicios [55] [52].

Independiente de la implementación que se utilice, las definiciones y librerías de JAX-RS tienen configuraciones de seguridad que protegen los servicios allí implementados de ataques maliciosos, aunque no hay certeza sobre los niveles de seguridad que estos pueden ofrecer, por tanto, tampoco se puede tener una información completa a la hora de decidir por el tipo de servicio que se debe desarrollar frente a una determinada necesidad.

Para la creación de un servicio REST es necesario seguir los siguientes principios [52] [56] (Documento):

- **Métodos HTTP:** GET, POST, PUT y DELETE.
- **Sin estado:** ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes.

- **Identificadores de recursos uniforme (URI):** cada recurso es accedido únicamente a través de su URI.
- **Hipermedios:** permite transferir los formatos HTML, JSON y XML.

2.4 Patrones arquitectónicos

Los patrones arquitectónicos son una descripción de un problema particular y recurrente de diseño, que aparece en contextos de diseño específico, y presenta un esquema genérico demostrado con éxito para su solución. El esquema de solución se especifica mediante la descripción de los componentes que la constituyen, sus responsabilidades y desarrollos, así como también la forma como estos colaboran entre sí [57].

Un patrón arquitectónico es una descripción de una organización del sistema, tal como una organización cliente-servidor o una arquitectura por capas. Los patrones arquitectónicos captan la esencia de una arquitectura que se usó en diferentes sistemas de software [51].

En la presente investigación se utiliza el patrón arquitectónico MVC, pues se encarga de separar la lógica de negocio de la interfaz de usuario.

Patrones de diseño a utilizar

Como antes se ha mencionado los patrones de diseño es una solución a un problema en un contexto en particular. Estos patrones a su vez son recurrentes por lo que hace que una solución a otras situaciones. Permitiendo que se adapte a la variante particular del problema que se quiere solucionar.

Patrones GRASP

Los patrones GRASP (Patrones Generales de Software para Asignar Responsabilidades en sus siglas en inglés General Responsibility Assignment Software Patterns) describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones [58].

De los patrones GRASP a utilizar es el controlador, el cual es un objeto que no pertenece a la interfaz de usuario, responsable de recibir o manejar un evento del sistema. Además, define el método para la operación del sistema.

Patrones GoF

Los patrones GoF fueron introducido popularmente a la industria del software con el libro Design Patterns: Elements of reusable object oriented programming escrito por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides en 1995. Estos introducen estructuras de clases e interfaces que solucionan un problema general, pensado para que las aplicaciones robustas sean escalables y flexibles, además rápidas de desarrollar y diseñar.

Los autores lo enfocan en distintos campos [59] [60]:

- **Patrones de diseño estructurales:** se enfocan en el proceso de instanciación y esto proporciona la ventaja de depender más en la composición de objetos que la herencia de clases.
- **Patrones de diseño de comportamiento:** Se enfocan en facilitar el diseño, al identificar formas sencillas de relacionar entidades. Estos patrones describen formas de componer objetos en caso de agregar nuevas funcionalidades a dichos objetos.
- **Patrones de diseño de creacionales:** Los patrones de diseño de comportamiento se enfocan en la comunicación de los objetos de las clases.

De los patrones GoF a utilizar en la presente investigación es el patrón de diseño de comportamiento Facade este patrón provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema.

2.5 Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que proveerá el sistema, de la manera en que éste reaccionará a entradas particulares. En algunos casos, los requerimientos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer.

En el presente epígrafe se enumeran los requisitos funcionales definidos en la investigación:

- 1) Validar campo de texto contraseña.
- 2) Establecer la conexión con los repositorios de patrones de diseño.

- 3) Introducir el problema de diseño de recursos educativos que se desea resolver de recursos educativos.
- 4) Realizar el análisis comparativo de los textos de los problemas para la recomendación de los catálogos.
- 5) Realizar el análisis comparativo de los textos de los problemas para la recomendación de los lenguajes.
- 6) Listar los patrones de diseño obtenidos en el análisis comparativo.
- 7) Listar los lenguajes de patrones de diseño obtenidos en el análisis comparativo.
- 8) Listar los catálogos de patrones de diseño obtenidos en el análisis comparativo.
- 9) Obtener el índice de interacción de los usuarios para el indicador recomendación.
- 10) Obtener el índice de interacción de los usuarios para el indicador descarga.
- 11) Obtener el índice de interacción de los usuarios para el indicador evaluación.
- 12) Calcular el nivel de satisfacción a través de los parámetros de medición de la calidad.
- 13) Ordenar el resultado de la búsqueda de los catálogos, lenguajes y patrones independientes teniendo en cuenta los parámetros de medición de la calidad.
- 14) Mostrar la procedencia de los patrones recomendados.

2.6 Requisitos no funcionales

Los requisitos no funcionales son restricciones de las funciones o servicios ofrecidos por el sistema. Estos requisitos incluyen restricciones de tiempo, sobre el proceso de desarrollo y prestaciones que deben tener el hardware y software con los cuales se desarrolla la aplicación, además de especificar el comportamiento del producto.

Para determinar los tipos de requisitos que se identificaron en la investigación, se utilizó el modelo ISO 25010, el cual determina las características de calidad que se toman en cuenta para evaluar las propiedades de un producto de software determinado [61].

Además, uno de los tipos de requisitos que se quiere sobresaltar son los requisitos de usabilidad. Para ello se utilizó el modelo ISO 25010 y por los principios planteados por Jakob Nielsen.

Requisitos de Seguridad.

- El sistema debe ser seguro ante ataques externos e internos. El sistema basa su seguridad en la autenticación de usuarios, donde se le asigna a los usuarios de la aplicación un usuario y contraseña.

Requisitos de Fiabilidad.

- La aplicación debe mostrar resultados que satisfagan las necesidades de los usuarios en cuanto a los patrones de diseño, para credibilidad y seguridad del mismo.
- El servidor web debe constar con los requisitos de hardware suficientes para que todos los usuarios, puedan acceder con rapidez. Es decir que la aplicación esté disponible el mayor tiempo posible.

Requisitos de Disponibilidad.

- El sistema debe poder ser accedido de forma concurrente por un elevado número de usuarios.

Requisitos de Escalabilidad.

- El sistema debe poder ampliarse con nuevas funcionalidades.

Para que el sistema funcione requiere de requisitos de hardware tales como:

Red LAN: El sistema para establecer la conexión con los repositorios de patrones de diseño se basa en la arquitectura SOA por lo que necesita un medio de comunicación. La Red LAN servirá para comunicar el sistema (aplicación de escritorio) con los repositorios.

PC: Memoria RAM 512 MB, 1GB Disponible de Disco Duro, Procesador 1 Ghz de velocidad.

2.7 Historias de usuario

En el presente epígrafe se muestran algunas de las historias de usuarios correspondientes a los requisitos definidos en la investigación:

TABLA 2. HU VALIDAR CAMPO DE TEXTO CONTRASEÑA

Número: 1	Nombre del requisito: Validar campo de texto contraseña	
Programador: Naylin Corrales Sánchez		Iteración Asignada: 1era
Prioridad: Alta		Tiempo Estimado: 1 día
Riesgo en Desarrollo: N/A		Tiempo Real: 2 horas
<p>Descripción:</p> <p>1- Objetivo:</p> <ul style="list-style-type: none"> - Permitir validar el campo de texto de la contraseña. <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para validar campo de texto hay que:</p> <ul style="list-style-type: none"> - Insertar la dirección ip, usuario y contraseña. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - Cuando inserta la contraseña de conexión no le muestra al usuario la contraseña. 		
Observaciones:		
Prototipo de interfaz:		

TABLA 3. HU ESTABLECER LA CONEXIÓN CON LOS REPOSITORIOS DE PATRONES DE DISEÑO

Número: 2	Nombre del requisito: Establecer la conexión con los repositorios de patrones de diseño	
Programador: Naylin Corrales Sánchez	Iteración Asignada: 1era	
Prioridad: Alta	Tiempo Estimado: 16 días	
Riesgo en Desarrollo: N/A	Tiempo Real: 15 días	
<p>Descripción:</p> <p>1- Objetivo:</p> <ul style="list-style-type: none"> - Permitir establecer la conexión con los repositorios de los patrones de diseño. <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para establecer la conexión con los repositorios de patrones de diseño hay que:</p> <ul style="list-style-type: none"> - Introducir el usuario, la contraseña y la dirección IP. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - Cuando el usuario selecciona la opción “Aceptar”, se muestra la interfaz de recomendación. El sistema internamente establece la conexión con los repositorios de patrones de diseño. - Cuando el usuario selecciona la opción “Cancelar”, se cierra la aplicación. 		
Observaciones:		
Prototipo de interfaz:		



TABLA 3. HU INTRODUCIR EL PROBLEMA DE RECURSOS EDUCATIVOS

Número: 3	Nombre del requisito: Introducir el problema de recursos educativos	
Programador: Naylin Sánchez	Corrales	Iteración Asignada: 1era
Prioridad: Alta		Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A		Tiempo Real: 2 días
<p>Descripción:</p> <p>1- Objetivo:</p> <ul style="list-style-type: none"> - Permitir introducir el problema de recursos educativos. <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para introducir el problema de recursos educativos hay que:</p>		

- Establecer la conexión con los repositorios de patrones de diseño, introduciendo el usuario, contraseña y dirección ip.

3- Comportamientos válidos y no válidos (flujo central y alternos):

- El campo de introducir el problema de diseño de recursos educativos es obligatorio.
- Introducir problema: campo de texto que permite caracteres alfabéticos y tiene un máximo de hasta 250 caracteres.

4- Flujo de la acción a realizar:

- El sistema debe permitir introducir el problema de diseño de recursos educativos para la recomendación de lenguajes, patrones pertenecientes a catálogos y patrones de diseño independientes.
- Cuando el usuario introduce el problema de diseño de recursos educativos y seleccione la opción "Recomendación", se le muestra el listado de lenguajes, patrones pertenecientes a catálogos y patrones de diseño independientes que pudieran dar respuesta a ese problema.
- Si el problema es incorrecto se le mostrará un mensaje al usuario para notificarle que el problema es incorrecto, y se le dará la posibilidad al usuario de introducir nuevamente el problema.

Observaciones:

Prototipo de interfaz:

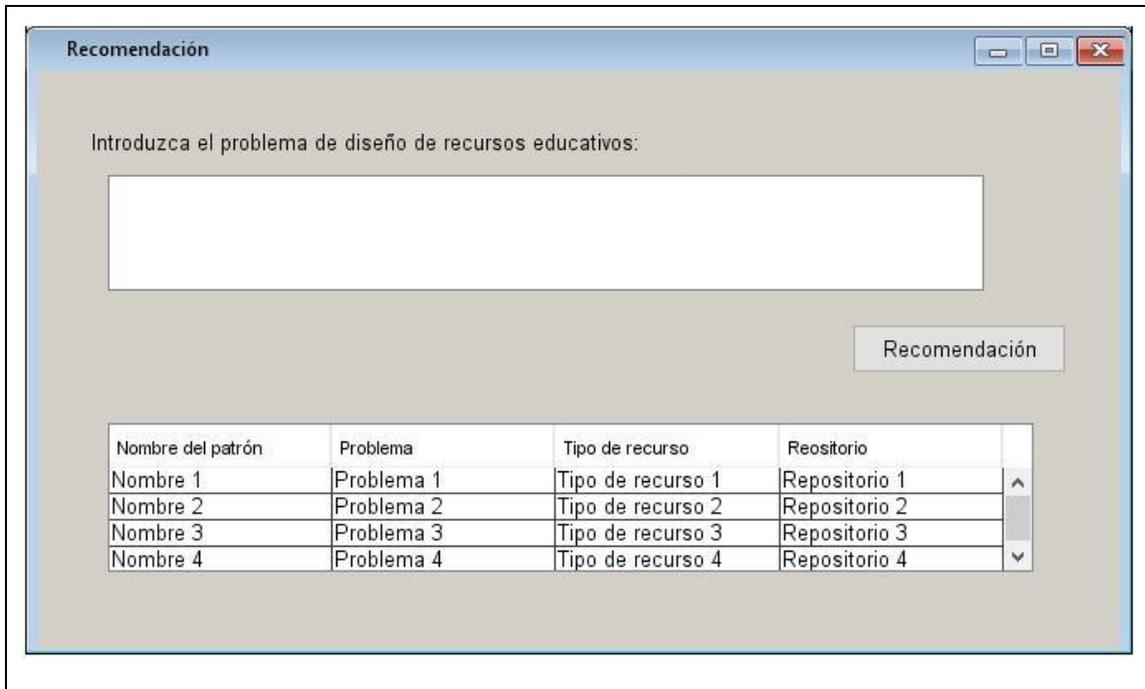


TABLA 4. HU LISTAR LOS PATRONES DE DISEÑO OBTENIDOS EN EL ANÁLISIS COMPARATIVO

Número: 6	Nombre del requisito: Listar los patrones de diseño obtenidos en el análisis comparativo	
Programador: Naylin Corrales Sánchez	Iteración Asignada: 1era	
Prioridad: Alta	Tiempo Estimado: 3 días	
Riesgo en Desarrollo: N/A	Tiempo Real: 2 días	
Descripción: <ul style="list-style-type: none"> - Una vez el usuario haya introducido el problema a resolver se le debe mostrar un listado de los lenguajes, patrones pertenecientes a catálogos y patrones de diseño independientes que pudiera utilizar. 		
1- Objetivo:		

- Permitir listar los patrones de diseño en el sistema.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para listar los patrones de diseño hay que:

- Establecer la conexión con los repositorios de patrones de diseño, introduciendo la dirección ip, usuario y contraseña.
- Introducir el problema de diseño de recursos educativos que se desea resolver.

3- Flujo de la acción a realizar:

- Cuando el usuario selecciona la opción "Recomendación" aparecen los patrones de diseño que dan respuesta al problema introducido por él. Además, el usuario tiene la posibilidad de ver los elementos principales con los que están compuestos dichos patrones.

Observaciones:

Prototipo de interfaz:

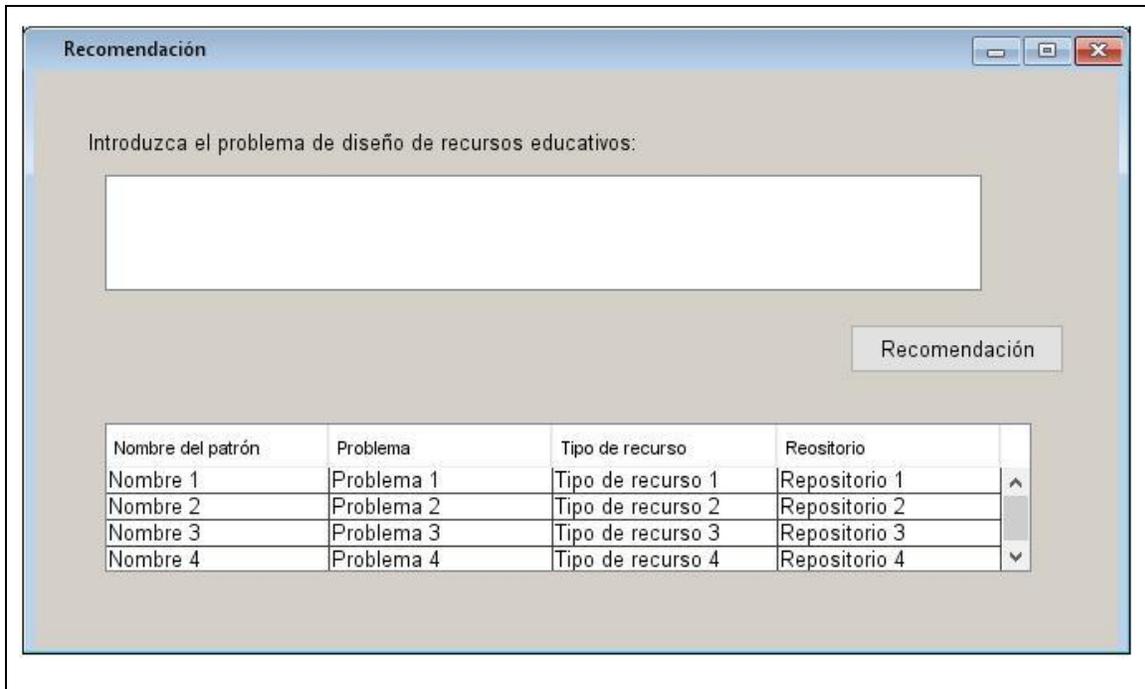


TABLA 5. HU LISTAR LOS LENGUAJES DE PATRONES DE DISEÑO OBTENIDOS EN EL ANÁLISIS COMPARATIVO

Número: 7	Nombre del requisito: Listar los lenguajes de patrones de diseño obtenidos en el análisis comparativo		
Programador: Naylin Sánchez	Corrales	Iteración Asignada: 1era	
Prioridad: Alta		Tiempo Estimado: 3 días	
Riesgo en Desarrollo: N/A		Tiempo Real: 2 días	
Descripción: <ul style="list-style-type: none"> - Una vez el usuario haya introducido el problema a resolver se le debe mostrar un listado de los lenguajes de patrones de diseño que se pudieran utilizar. 			
1- Objetivo:			

- Permitir listar los lenguajes de patrones de diseño en el sistema.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para listar lenguajes de patrones de diseño hay que:

- Establecer la conexión con los repositorios de patrones de diseño, introduciendo la dirección ip, usuario y contraseña.
- Introducir el problema que se desea resolver.

3- Flujo de la acción a realizar:

- Cuando el usuario selecciona la opción “Recomendación” aparece el listado los lenguajes de patrones de diseño que dan respuesta al problema introducido por él. Además, el usuario tiene la posibilidad de ver los elementos principales con los que están compuestos dichos lenguajes.

Observaciones:

Prototipo de interfaz:

Recomendación

Introduzca el problema de diseño de recursos educativos:

Recomendación

Nombre del patrón	Problema	Tipo de recurso	Repositorio
Nombre 1	Problema 1	Tipo de recurso 1	Repositorio 1
Nombre 2	Problema 2	Tipo de recurso 2	Repositorio 2
Nombre 3	Problema 3	Tipo de recurso 3	Repositorio 3
Nombre 4	Problema 4	Tipo de recurso 4	Repositorio 4

TABLA 6. HU LISTAR LOS CATÁLOGOS DE PATRONES DE DISEÑO

Número: 8	Nombre del requisito: Listar los catálogos de patrones de diseño obtenidos en el análisis comparativo	
Programador: Naylin Corrales Sánchez	Iteración Asignada: 1era	
Prioridad: Alta	Tiempo Estimado: 3 días	
Riesgo en Desarrollo: N/A	Tiempo Real: 2 días	
<p>Descripción:</p> <ul style="list-style-type: none"> - Una vez el usuario haya introducido el problema a resolver se le debe mostrar un listado de los catálogos de patrones de diseño que pudiera utilizar. <p>1- Objetivo:</p> <ul style="list-style-type: none"> - Permitir listar los catálogos de patrones de diseño en el sistema. <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para listar los patrones de diseño hay que:</p> <ul style="list-style-type: none"> - Establecer la conexión con los repositorios de patrones de diseño, introduciendo la dirección ip, usuario y contraseña. - Introducir el problema que se desea resolver. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - Cuando el usuario selecciona la opción “Recomendación” aparece el listado los patrones pertenecientes a catálogos que dan respuesta al problema 		

introducido por él. Además, el usuario tiene la posibilidad de ver los elementos principales con los que están compuestos dichos patrones.

Observaciones:

Prototipo de interfaz:

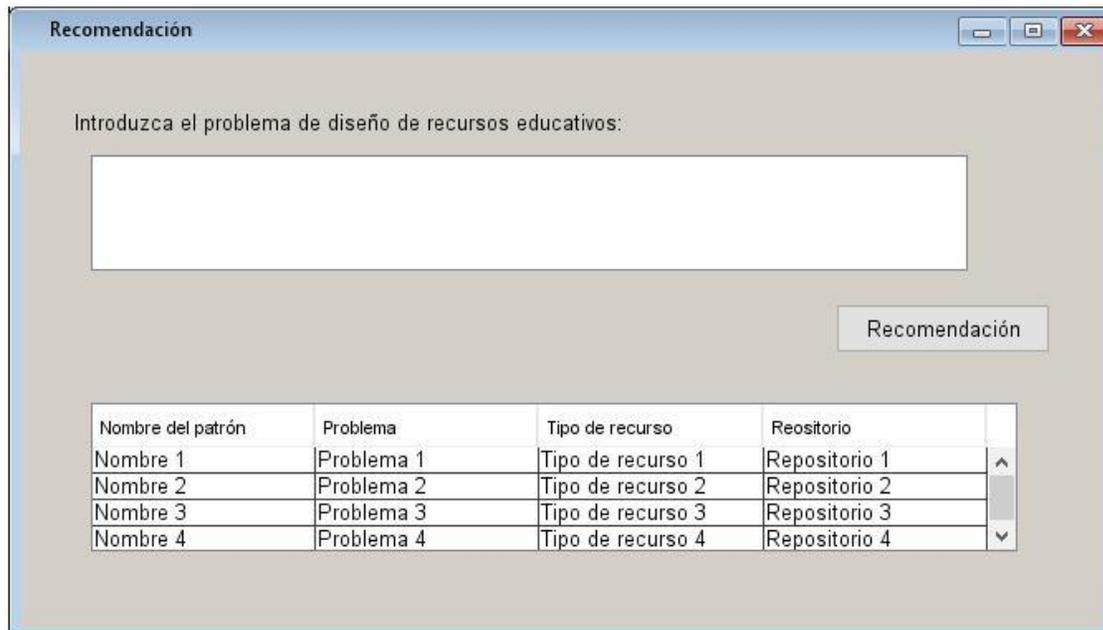


TABLA 7. MOSTRAR LA PROCEDENCIA DE LOS PATRONES RECOMENDADOS

Número: 14	Nombre del requisito: Mostrar la procedencia de los patrones recomendados.		
Programador: Naylin Sánchez	Corrales	Iteración Asignada: 1era	
Prioridad: Alta		Tiempo Estimado: 10 días	
Riesgo en Desarrollo: N/A		Tiempo Real: 8 días	
Descripción:			

1- Objetivo:

- Permitir mostrar al usuario la procedencia del lenguaje, catálogo o patrón que responda el problema introducido.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para mostrar la procedencia hay que:

- Establecer la conexión con los repositorios de patrones de diseño, introduciendo la dirección ip, usuario y contraseña.
- Introducir el problema que se desea resolver.

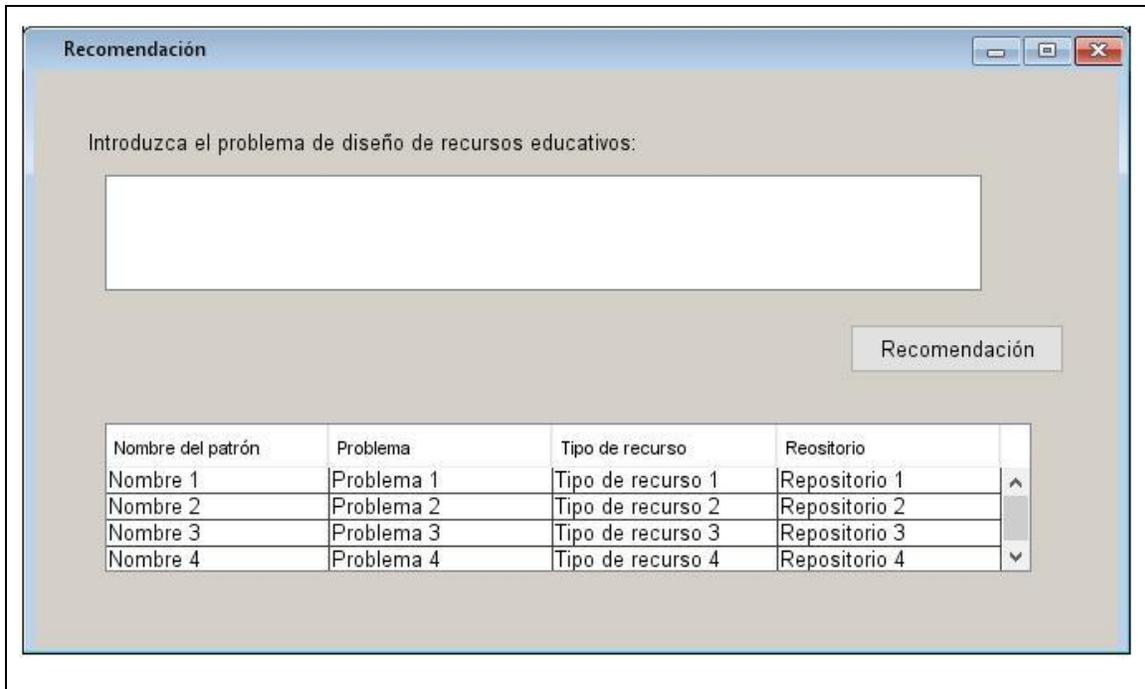
3- Comportamientos válidos y no válidos (flujo central y alternos):

- Debe hacerse cuando el usuario introduzca el problema que desea resolver y selecciona la opción "Recomendación".

Flujo de la acción a realizar:

- Una vez que se realizó la conexión a los repositorios de patrones de diseño y que el usuario introduce el problema de diseño de recursos educativos le muestra una tabla los patrones pertenecientes a catálogos, los lenguajes y los patrones independientes que responden al problema introducido y el repositorio donde se obtuvo esos patrones.

Observaciones:**Prototipo de interfaz:**



2.8 Conclusiones parciales

A través de la selección del patrón arquitectónico MVC, los patrones de diseño GRASP y GoF, se puede realizar la implementación del sistema posibilitando una organización adecuada del código de la aplicación. Además, mediante el servicio web REST se pudo establecer la conexión a los repositorios de patrones de diseño, permitiendo el consumo de servicios web para determinar el nivel de satisfacción de los usuarios al interactuar con determinados patrones.

Capítulo 3. Implementación y validación del sistema de recomendación

En el presente capítulo se detalla los métodos implementados para darle cumplimiento al problema trazado en la investigación. Además, se evidencian las pruebas empleadas para validar el sistema desarrollado como la de caja blanca, la de caja negra y la técnica de ladov.

3.1 Implementación

En ciencias de la computación, una implementación es la realización de una especificación técnica o algoritmos como un programa, componente de software u otro sistema de cómputo. Muchas implementaciones son dadas según a una especificación o un estándar [16].

La etapa de implementación consistirá en la codificación del sistema de recomendación a partir de los artefactos obtenidos en la etapa de diseño, en este caso las historias de usuario.

3.1.1 Estándares de codificación

Los estándares de codificación son reglas o patrones de escritura del código de programación que permiten tener una programación homogénea y que regula la realización de ciertos procesos, de esta forma el sistema resulta fácil de entender y mantener, independientemente de su autor.

En la implementación del sistema de recomendación de patrones de diseño de recursos educativos se utilizaron varios estándares para favorecer la legibilidad y organización del código.

Convenciones de codificación para la nomenclatura:

- Las **variables de clase o método y las instancias** deben declararse en formato *camelCase*, empezar por letra minúscula y, si se trata de palabras concatenadas, la primera letra de las siguientes palabras en mayúscula.

```
public class Catalogo {  
    private int id;  
    private String nombre;  
    private String clasificacion;  
    private String descripcionProblema;
```

- Las **variables constantes** se deben declarar totalmente en mayúsculas, separando las palabras con el guión bajo (“_”) si se trata de una palabra compuesta.
- Las **clases** deben tener la primera letra mayúscula al igual que la primera letra de las siguientes palabras si se trata de una palabra compuesta. Se recomienda que los identificadores de las clases sean sustantivos.

```
public class Catalogo {

public class Conjunto {

public class Lenguajes {

public class Patron {
```

3.1.2 Funcionalidades implementadas

En el presente epígrafe se aborda sobre los métodos más importantes implementados.

```
private void botonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        if(usuario.getText().equals("") ||jPasswordField1.getText().equals("")){
            JOptionPane.showMessageDialog(this, "Introduzca los datos", "Información", JOptionPane.INFORMATION_MESSAGE);
        }else{
            Connection dbConnection = null;
            Class.forName("org.postgresql.Driver");
            String direccion= jtextfil.getText();
            String db_address = "jdbc:postgresql://" + direccion + ":5432/pattern";
            //JOptionPane.showMessageDialog(null, db_address);
            dbConnection = DriverManager.getConnection(db_address,usuario.getText(),jPasswordField1.getText());
            dispose();
            problema D = new problema(null, true, c, dbConnection);
            D.setVisible(true);
        }catch (Exception e) {
            System.out.println(e.getMessage());
            JOptionPane.showMessageDialog(this, "Error en la conexión, introduzca los datos correctamente", "Error",
        }
    }
}
```

ILUSTRACIÓN 8. CÓDIGO PARA ESTABLECER LA CONEXIÓN

```

public int similitudEntreDosCadenas(String s, String l) {
    int resulta = 0;
    int contador = 0;
    Stemm_es si = new Stemm_es();
    LinkedList<String> arreglo = eliminarPala(s);
    LinkedList<String> arre = eliminarPala(l);
    LinkedList<String> lis = new LinkedList<String>();
    LinkedList<String> lis1 = new LinkedList<String>();
    for (int i = 0; i < arreglo.size(); i++) {
        String m = si.stemm(arreglo.get(i));
        lis.add(m);
    }
    for (int i = 0; i < arre.size(); i++) {
        String mi = si.stemm(arre.get(i));
        lis1.add(mi);
    }
    if (lis.size() == 0 || lis1.size() == 0) {
        return -1;
    } else {
        for (int i = 0; i < lis.size(); i++) {
            for (int j = 0; j < lis1.size(); j++) {
                if (lis.get(i).equalsIgnoreCase(lis1.get(j))) {
                    contador++;
                }
            }
        }
    }
}

```

ILUSTRACIÓN 9. CÓDIGO PARA LA SIMILITUD ENTRE CADENAS

```

public int Semejanza(String proble, String caso) {
    int seme = 0;
    LinkedList<String> l = extraerNgramas(proble);
    LinkedList<String> li = extraerNgramas(caso);
    Conjunto conj = new Conjunto(l);
    Conjunto conji = new Conjunto(li);
    int inter = conj.Inter(l, li);
    Conjunto union = conj.Union(conji);
    if ((float) inter / union.getS().length >= 0.5) {
        seme = 1;
    }

    return seme;
}

```

ILUSTRACIÓN 10. CÓDIGO PARA CUANDO LAS CADENAS SON SEMEJANTES

Una vez finalizada la implementación del sistema y antes de realizar la evaluación de los objetivos del proyecto, es necesario realizar pruebas a los algoritmos para comprobar su correcto funcionamiento.

3.2 Pruebas

Las pruebas de software son importantes porque aseguran el correcto cumplimiento de la funcionalidad del producto, ayudan a ganar confianza, confirman la fiabilidad del uso y previenen defectos en producción, lo cual tiene un impacto económico positivo en la empresa en cuestión [62] [63].

El propósito de las pruebas es [62]:

- Detectar la mayor cantidad de errores posibles.
- Ayudar a los administradores a la toma de decisiones.
- Evaluar la calidad.
- Verificar la corrección del producto.
- Asegurar la calidad.

- Alcanzar mayor calidad en el software.
- Cambios en la tecnología.

Reducción de costos y riesgos.

Existen tres tipos de pruebas [63]:

- **Las pruebas funcionales:** Se basa en las funcionalidades del sistema, donde se describen en la especificación de requisitos, es decir, lo que hace que le sistema. Las pruebas funcionales suelen estar asociadas a las técnicas de diseño de pruebas de caja negra, pues tienen en cuenta el comportamiento externo del software.
- **Las pruebas no funcionales:** tienen en cuenta el comportamiento externo del software, es decir cómo funciona el sistema, y suelen utilizar las técnicas de diseño de caja negra. Al igual que las características funcionales, las características no funcionales tienen que estar definidas en las especificaciones del software.
- **Las pruebas estructurales:** permiten medir la totalidad de las pruebas mediante la evolución de tipo de estructura. En estas pruebas se aplica las técnicas de diseño de caja blanca.

En la presente investigación se utiliza el método de caja negra, aplicando la técnica de partición de equivalencia, que permite examinar los valores válidos o inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genético y además se utiliza la prueba de caja blanca a todo el código de la aplicación desde que se empezó a implementar el sistema.

Prueba de Caja Blanca

Las técnicas de caja blanca, examinan la parte interna del programa, siempre se está observando el código, y los casos de prueba están basados en la estructura interna del programa (Ilustración 9). Por ello, la implementación de estas pruebas depende de la disponibilidad del código fuente [62] [63].

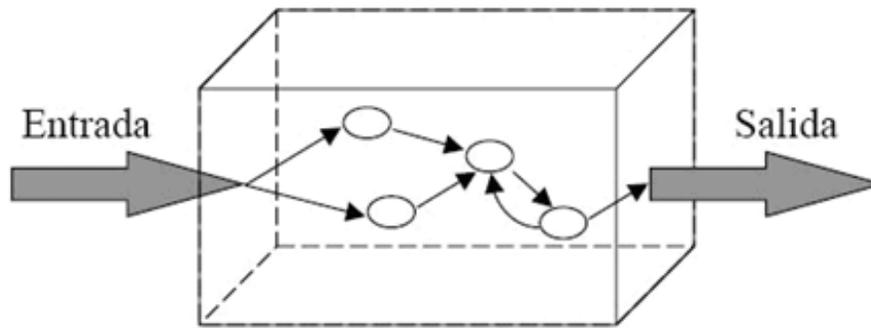


ILUSTRACIÓN 11. PRUEBA DE CAJA BLANCA

Esta prueba se realiza durante toda la etapa de implementación del sistema, para poder probar y ejecutar el código generado.

Prueba de Caja Negra

La técnica de caja negra, se enfoca en probar el sistema sin tomar en cuenta la estructura interna del mismo, su objetivo es validar que las salidas sean las esperadas (Ilustración 10) [62] [63].



ILUSTRACIÓN 12. PRUEBA DE CAJA NEGRA

Para el empleo de esta prueba se utiliza una de las técnicas propuesta por ella la prueba de partición equivalente.

Prueba de partición equivalente

De las técnicas que propone esta prueba se utiliza la técnica partición de equivalencia la cual se encarga de dividir los posibles valores en clases, valores de entrada y valores de salida.

Se agrupan todos los valores para los cuales se espera que el programa tenga un comportamiento común (rango de valores), y esa es una clase de equivalencia, existen clases de equivalencia válidas y clases de equivalencias inválidas [62].

Descripción de variables

En la siguiente tabla se refleja las variables con que cuenta el sistema de recomendación.

TABLA 8. DESCRIPCIÓN DE VARIABLES

No.	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Usuario	Campo de texto	No	Campo de carácter obligatorio que representa el usuario, puede contener letras, comenzando con mayúsculas de 3 hasta 255 caracteres de longitud. Ejemplo: ncorrales.
2	Contraseña	Campo de texto	No	Campo de carácter obligatorio que representa el usuario, puede contener letras y números.
3	Problema	Campo de texto	No	Campo de carácter obligatorio que representa el usuario, puede contener letras y números.

Casos de prueba

En el presente epígrafe se detallan algunos de los casos de prueba correspondientes a los requisitos funcionales identificados en la investigación.

Requisito: “Establecer la conexión con los repositorios de patrones de diseño”

Descripción general: El requisito comienza cuando el usuario ejecuta la aplicación, mostrándole una ventana con un formulario para que el usuario se autentique y culmina cuando se presiona el botón “Conectar”.

TABLA 9. CASO DE PRUEBA RF. ESTABLECER LA CONEXIÓN CON LOS REPOSITORIOS DE PATRONES DE DISEÑO

Escenario	Descripción	Usuario	Contraseña	Respuesta del sistema	Flujo central
EC 1.1 Ejecuta la aplicación	Muestra un formulario de conexión con repositorios	N/A	N/A	Muestra los datos que se deben introducir para establecer la conexión	Inicio
EC 1.2 Selecciona la opción “Conectar”	Introduce los datos y selecciona la opción “Conectar”	V	V	Valida los datos introducidos	Inicio/Conectar
EC 1.3 Existen campos vacíos o datos incorrectos	Existen campos vacíos o datos incorrectos	I	V	Muestra un mensaje informando cual es el campo incorrecto	Inicio/Conectar
		V	I		
EC 1.4 Selecciona la opción “Cancelar”	Se selecciona la opción “Cancelar”	N/A	N/A	Se cierra el sistema	Inicio/Cancelar

Requisito: “Introducir el problema de diseño de recursos educativos”

Descripción general: El requisito comienza cuando el usuario selecciona la opción “Conectar”, mostrándole una ventana para que el usuario introduzca el problema del recurso educativo y culmina cuando se presiona el botón “Recomendar”.

Condición de ejecución: El usuario debe estar autenticado en el sistema.

TABLA 10. CASO DE PRUEBA RF. INTRODUCIR EL PROBLEMA DE DISEÑO DE RECURSOS EDUCATIVOS

Escenario	Descripción	Problema	Respuesta del sistema	Flujo central
EC 1.1 Selecciona la opción “Conectar”	Muestra una ventana con la opción de introducir un campo de texto	N/A	Muestra los datos que se deben introducir para realizar la recomendación	Inicio/ Conectar
EC 1.2 Selecciona la opción “Recomendar”	Introduce los datos y selecciona la opción “Recomendar”	V	Valida los datos introducidos	Inicio/Conectar/ Recomendar
EC 1.3 Existen campos vacíos o datos incorrectos	Existen campos vacíos o datos incorrectos	I	Muestra un mensaje informando cual es el campo incorrecto	Inicio/Conectar
EC 1.4 Selecciona la opción “Cancelar”	Se selecciona la opción “Cancelar”	N/A	Se cierra el sistema	Inicio/Conectar/ Cancelar

Requisito: “Mostrar la procedencia de los patrones recomendados”

Descripción general: El requisito comienza cuando el usuario selecciona la opción “Acceder”, mostrándole en una ventana de forma detallada los parámetros del tipo de recurso

seleccionado (lenguaje, patrón perteneciente a un catálogo o un patrón independiente) y culmina cuando se presiona el botón “Regresar”.

Condiciones de ejecución:

- El usuario debe estar autenticado en el sistema.
- El problema introducido debe estar correcto.

TABLA 11. CASO DE PRUEBA RF. MOSTRAR LA PROCEDENCIA DE LOS PATRONES RECOMENDADOS

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Selecciona la opción “Recomendar”	Muestra el listado de la recomendación	Muestra el listado de la recomendación	Inicio/Conectar/ Recomendar
EC 1.2 Selecciona la opción “Acceder”	Se selecciona la opción “Acceder” que se encuentra en la última columna de la tabla	Muestra los detalles específicos del tipo de recurso seleccionado	Inicio/Conectar/ Recomendar/Acceder
EC 1.4 Selecciona la opción “Regresar”	Selecciona la opción “Regresar”	Regresa al listado de la recomendación	Inicio/Conectar/Recomendar/ Acceder/Regresar

Clasificación de las no conformidades

Una no conformidad es un fallo en el sistema de gestión de la calidad que puede producirse por varias razones: no alcanzar el nivel de aceptación establecido en un determinado indicador y errores en la documentación del sistema. Se trata de una desviación entre lo que hay escrito y lo que ha ocurrido. Este fallo queda registrado en un informe y se establecen las acciones preventivas y correctivas necesarias para arreglar lo que no funcione y evitar que vuelva a ocurrir. Las mismas se clasifican de acuerdo al nivel de importancia en [16] [64]:

- **Significativas:** Son aquellas que afectan la calidad del producto o servicio de manera visible, impidiendo o no el cumplimiento de algún requisito.
- **No significativas:** Son aquellas que resultan menos visibles, que no atentan contra el cumplimiento de algún requisito.
- **Recomendaciones:** Son aquellas que quedan en función de la apreciación del probador para oportunidades de mejoras del producto o servicio.

Para corregir los errores encontrados durante la implementación del sistema se realizaron dos iteraciones de pruebas, en las cuales se detectaron un conjunto de no conformidades. En la primera iteración se detectaron dos no conformidades, clasificadas como significativas resolviéndose ambas y en la segunda iteración se detectó una no conformidad, clasificada como no significativa a la cual se le dio respuesta.

3.3 Técnica IADOV

Para evaluar la satisfacción de los usuarios en cuanto al uso del sistema de recomendación, se recogió el criterio de usuarios, utilizando la Técnica de IADOV. Mediante dicha técnica se determina el nivel de satisfacción individual y grupal a partir de una encuesta elaborada compuesta de cinco preguntas: tres cerradas y dos abiertas [65] [66] [67], en la presente investigación, se aplicó a varios de los diseñadores de recursos educativos.

Una vez establecidas las preguntas se conforma el cuadro lógico de IADOV y el número resultante de la interrelación de las tres preguntas indica la posición de los sujetos en la escala de satisfacción.

TABLA 12. CUADRO LÓGICO DE IADOV

Pregunta 6. Marque con una X cuan satisfecho está con el procedimiento propuesto para la recomendación de patrones de diseño de recursos educativos.	Pregunta 4. ¿Considera usted que la utilización del tratamiento de agrupaciones de los patrones de diseño contribuye a una mejor recomendación?		
	Sí	No sé	No

	Pregunta 5. ¿Considera usted que teniendo en cuenta la calidad percibida por el usuario contribuye a una mejor recomendación de patrones de diseño?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho.	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta.	2	2	3	2	3	3	6	3	6
Me es indiferente.	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta.	6	3	6	3	4	4	3	4	4
No me gusta en lo absoluto.	6	6	6	6	4	4	6	4	5
No sé qué decir.	2	3	6	3	3	3	6	3	4

Para obtener el índice de satisfacción grupal se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y -1 de la siguiente forma:

1. Clara satisfacción.
2. Más satisfecho que insatisfecho.
3. No definida.
4. Más insatisfecho que satisfecho.
5. Clara insatisfacción.
6. Contradictoria.

El nivel de satisfacción grupal (ISG) se calcula por la siguiente fórmula:

$$\text{ISG} = \frac{A (+1) + B (+0.5) + C (0) + D (-0.5) + E (-1)}{N}$$

ILUSTRACIÓN 13. FÓRMULA GRUPAL DE IADOV

Donde: A, B, C, D, E, representan el número de sujetos con índice individual 1; 2; 3 ó 6; 4;

5 respectivamente y N representa el número total de sujetos del grupo.

Las siguientes categorías permiten conocer el nivel de satisfacción grupal:

- Insatisfacción: desde (-1) hasta (-0,5).
- Contradictorio: desde (-0,49) hasta (+0,49).
- Satisfacción: desde (+0,5) hasta (1).

Para medir el grado de satisfacción se tomó una muestra de 12 usuarios teniendo en cuenta su experiencia en la creación de recursos educativos. Después de realizar el análisis del cuestionario aplicado se obtuvo el siguiente resultado:

TABLA 13. NIVELES DE SATISFACCIÓN

Total de encuestados	12
Clara satisfacción	7
Más satisfechos que insatisfechos	3
No definida	1
Más insatisfechos que satisfechos	1
Clara insatisfacción	0

Contradictoria	0
----------------	---

Una vez obtenidos los resultados, se puede calcular el ISG a partir de los valores que toman las variables y que pueden apreciarse en la siguiente tabla:

TABLA 14. RESULTADOS DE LOS ENCUESTADOS

Variable	Valor
A	7
B	3
C	1
D	1
E	0

Calculando entonces el ISG quedaría:

$$\text{ISG} = \frac{7(+1) + 3(+0.5) + 1(0) + (-0.5) + 0(-1)}{12}$$

$$\text{ISG} = \frac{8}{12}$$

$$\text{ISG} = 0.66$$

El resultado obtenido se enmarca en el intervalo de 0.5 a 1, por lo tanto, el nivel de satisfacción de la muestra es: Satisfactorio.

3.4 Conclusiones parciales

A partir del empleo de las pruebas de caja blanca, la de caja negra y dentro de esta última aplicando la técnica de partición equivalente, permitió validar el Sistema de recomendación de

patrones de diseño de recursos educativos. El uso de la técnica de ladov, permitió evaluar el nivel de satisfacción de los usuarios con respecto al sistema demostrando resultados satisfactorios.

Conclusiones Generales

Con el desarrollo de la presente investigación, se dio cumplimiento a los objetivos propuestos inicialmente, por lo que se puede arribar a las siguientes conclusiones:

- El análisis de los distintos tipos de sistemas de recomendación permitió seleccionar la recomendación híbrida entre la colaborativa y basada en conocimiento, lo que demostró realizar la recomendación de una manera más adecuada a las necesidades de los diseñadores de recursos educativos.
- Se logró realizar el diseño del sistema utilizando el patrón arquitectónico MVC y, además se obtuvo como resultado un diseño visual más afín a la estrategia marcaria de la UCI, lo que propició el cumplimiento de los requisitos establecidos por el cliente.
- Se implementó el Sistema para recomendar patrones de diseño de recursos educativos basado en el tratamiento de agrupaciones y la calidad percibida por los usuarios, utilizando REST para consumir los servicios necesarios, obtenidos de los distintos repositorios de patrones.
- Se validó el sistema a través de las pruebas de caja blanca y caja negra, donde se comprobó el correcto funcionamiento del sistema de recomendación. Y el empleo de la técnica de ladov evidenció una clara satisfacción por parte de los usuarios encuestados.

Recomendaciones

Para la continuidad del trabajo desarrollado la autora recomienda:

- Incluir, al sistema propuesto, funcionalidades que permitan crear lenguajes a partir de a partir de la agrupación de patrones de diseño de recursos educativos.

Referencias

1. UNESCO. *Las TIC en la educación*. 2017 14/10/2017]; Available from: <http://es.unesco.org/themes/tic-educacion>.
2. UNESCO. *Comunicación e información*. 2017 14/10/2017]; Available from: <http://www.unesco.org/new/es/communication-and-information/access-to-knowledge/open-educational-resources>.
3. Educativas, P. *¿Qué son los repositorios digitales?* 2014 17/01/2018]; Available from: <http://unid.edu.mx/tecnologiaeinnovacion/967-repositorios-digitales-html>.
4. Morgado, E.M., et al., *Desarrollo de competencias a través de objetos de aprendizaje*. Revista de Educación a distancia, 2013. Num. 36: p. 19.
5. García, A.M., *Patrones de Diseño aplicados a la organización de repositorios de objetos de aprendizaje*. Revista de Educación a Distancia, 2009: p. 20.
6. HillsideGroup. *About patterns*. 2017 21/11/2017]; Available from: <http://hillside.net/patterns/about-patterns>.
7. Alexander, C., *A pattern language*. Vol. 2. 1977.
8. Marciszack, M., et al., *Implementación de patrones en la validación de modelos conceptuales*. 2015.
9. Jiménez, J.M.R., *Patrones pedagógicos en educación virtual*. 2009 p. 16.
10. Gómez, Y.A. and Y.M. Martínez, *Sistema de recomendación de patrones de diseño para Recursos Educativos Abiertos*. 2015, Universidad de las Ciencias Informáticas. p. 73.
11. Castillo, W.A. and A.B. Medina, *Aplicación de recursos didácticos para la mejora de la competencia*. 2017, Universidad nacional de San Agustín de Arequipa. p. 103.
12. González, M.L.C., *Recursos educativos TIC de información, colaboración y aprendizaje*. Revista de Medios y Educación 2011. Num. 39: p. 69-81.

13. Fernández, A.R.A., *Aplicación de materiales y recursos educativos "Vitudimalufi" para desarrollar capacidades matemáticas en los estudiantes del primer año*. 2015, Universidad Nacional de San Agustín de Arequipa. p. 63.
14. Jimenez-Torres, V.H., W. Tello-Borja, and J.I. Rios-Patiño, *Lenguajes de Patrones de Arquitectura de Software: Una Aproximación Al Estado del Arte*. 2014. Vol. 19, No. 4: p. 371-376.
15. Pressman, R.S., *Ingeniería del software. Un enfoque práctico*, ed. S. edición. 2010.
16. González, P.R.P. and C.F. Cabrera, *Módulo para la gestión de patrones de diseño de Recursos Educativos Abiertos*. 2015, Universidad de la Ciencias Informáticas. p. 123.
17. Educación. *Patrones de diseño aplicados al desarrollo de Objetos Digitales Educativos*. 2017 [27/09/2017]; Available from: <http://ares.cnice.mec.es/informes/21/index.htm>.
18. Rodríguez, P., et al., *Validación de un sistema inteligente de recomendación híbrida en federaciones de repositorios de Objetos de Aprendizaje*. 2013. Vol. 11, No. 3: p. 10.
19. Sust, E.B.U. and A.J.S. Cuevas, *Sistemas de recomendación semánticos: Una revisión del Estado del Arte*. 2017. Vol. 11, No. 2: p. 189-206
20. Ricci, F., et al., *Recommender Systems Handbook*. 2011.
21. Vekariya, V. and G.R. Kulkarni, *Hybrid Recommender systems: survey and experiments*. Vol. 2: p. 556-561.
22. Carrillo, G. and X. Ochoa, *Recomendación de objetos de aprendizaje basado en el perfil del usuario y la información de atención contextualizada* 2012, Universidad Oberta de Catalunya. p. 21.
23. Burke, R., *Hybrid Recommender Systems: Survey and Experiments*, in *Department of Information Systems and Decision Sciences* 2004, California State University. p. 29.
24. Pazzani, M.J. and D. Billsus, *Content-Based Recommendation Systems* 2007: p. 325-341.

25. Campos, L.M.d., et al., *Uso de conocimiento estructurado en un sistema de recomendación basado en contenido*. 2006: p. 6.
26. Barranco, M.J., L.G. Pérez, and L. Martínez, *Un Sistema de Recomendación Basado en Conocimiento con Información Lingüística Multigranular*. 2006: p. 645-664.
27. Torrejón, D.A.R. and J.M.M. Ramos, *Detección de plagio en documentos. Sistema externo monolingüe de altas prestaciones basado en n-gramas contextuales*. 2009: p. 9.
28. Schwartz, F.P., *Análise do discurso parlamentar por meio da técnica do processamento de linguagem natural: abordagem estatística e aprendizagem de máquina*, in *Departamento de Engenharia Elétrica*. 2018, Universidade de Brasilia. p. 82.
29. Chirichetti, L., *"EL criado pesado": La caracterización en la serie Águila Roja*. *Quaderns de Filologia: Estudis Lingüístics*, 2017. 22: p. 57-78.
30. Microsoft. *Introducción a la búsqueda federada*. 2010 [11/06/2018]; Available from: [https://docs.microsoft.com/es-es/previous-versions/office/developer/sharepoint-2010/cc806030\(v%3Doffice.14\)](https://docs.microsoft.com/es-es/previous-versions/office/developer/sharepoint-2010/cc806030(v%3Doffice.14)).
31. Montalvo, M.M., *Plataformas de descubrimiento en bibliotecas universitarias: evaluación de WorldCat en el contexto de una biblioteca caribeña*. 2017. Vol. 20: p. 17.
32. Marín, P.A.R., et al., *Modelo multi-agente para recomendación híbrida de objetos de aprendizaje*. 2013. No. 40: p. 96-110.
33. Castillo, E.J.A., L.G. Estrada, and V. Estrada, *Sistema de recomendación basado en k-*nn* para condiciones de incertidumbre en un Sistema Tutor Inteligente*. 2014. Vol. 45, No. 3: p. 25 - 30.
34. Lobo, C. and J. Fernando, *Slideshare*. 2015.
35. Sánchez, T.R., *Metodología de desarrollo para la Actividad productiva de la UCI*. 2015: p. 16.

36. Águilar, L.J. and M.F. Azuela, *Java 2. Manual de programación*.
37. Orrala, R.D.T., *Desarrollo de un sistema de ventas de artesanías de los grupos de interés prioritario afines al FENEDIF, utilizando referencias en sistemas de reconocimiento de voz*. 2017, Universidad Estatal Península de Santa Elena. p. 66.
38. Adan Oliva, H.F., et al., *Generador de código automático (GCA) en lenguaje Java a partir de con conjunto de instituciones en lenguaje natural*. Jóvenes Investigadores, 2014. Vol. 1.
39. Hernández, L.R.B., et al., *Sistema para el control del cumplimiento del proyecto educativo en la enseñanza superior cubana*. Revista Científica, 2016: p. 83-89.
40. Fragas, Y.S., D.M. Peña, and P.M.H. Alfonso, *Sistema automatizado para la gestión del mantenimiento de equipos (módulos administración y solisitud de servicio)*. Revista Ciencias Técnicas Agropecuarias, 2015. Vol. 24: p. 85-90.
41. Silberschatz, A., H.F. Korth, and S. Sudarshan, *Fundamentos de bases de datos*. 2002, España.
42. Ordóñez, M.P.Z., J.R.M. Ríos, and F.F.R. Castillo, *Administración de bases de datos con PostgreSQL*. 2017.
43. PostgreSQL. *Características, limitaciones y ventajas*. 2018 [17/01/2018]; Available from: <https://www.postgresql.org/>.
44. Otero, B.G., *El debate sobre la interoperabilidad informática en el derecho de autor comunitario*. 2013, Universidad de Santiago de Compostela. p. 393.
45. Daniel, A.P. and A.J. Betancourt, *Implementación de estándares de interoperabilidad en el repositorio de recursos educativos RHODA*. 2014, Universidad de las Ciencias Informáticas. p. 93.
46. Caro, M.F., J. Hernández, and J.A. Jiménez, *Diseño de un sistema de recomendación en repositorios de objetos de aprendizaje basado en la percepción del usuario: Caso RODAS*, in *Ciencia e Ingeniería Neogranadina*. 2011. p. 51-72.

47. Cañizares, R., *Repositorio de Recursos Educativos para las Instituciones de Educación Superior*. 2012.
48. Peña, A.D.R. and L.G.S. Rojas, *Arquitectura de software para el sistema de visualización médica Vismedic*. 8, 2016: p. 75-86.
49. Alicante, U.d. *Modelo vista controlador (MVC)*. 2017 [5/3/2018]; Available from: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>.
50. UAH, I.d.S. *Arquitectura Orientada a Servicios (SOA)*. 2015 [09/03/2018]; Available from: <https://ingenieriadelsoftwareuah2015.wordpress.com/2015/03/22/arquitectura-orientada-a-servicios-soa>.
51. Sommerville, I., *Ingeniería de software*. Novena Edición ed. 2011. 792.
52. Gonzáles, J.C.F. and J.A.R. Oviedo, *Seguridad de JAX-RS frente a ataques por inyección de código*. 2016, Universidad Católica de Colombia. p. Pág. 105.
53. Pintado, E.N.R. and G. Abigail, *Análisis, diseño y desarrollo de un prototipo de Endpoint para mejorar la producción de web services*. 2015, Univesidad de Guayaquil. p. 172.
54. Macas, L.N.H. and M.E.J. Acaro, *Estudio comparativo de los servicios web RESTfull Jersey y SOAP JAX-WS para el desarrollo de una aplicación android con Wikitude aplicada a la gestión de información geolocalizada del turismo de la provincia de Chimborazo*. 2016, Escuela superior politécnica de Chimborazo. p. 149.
55. Consortium, W.W. *Web of Services*. 2015 [13/03/2018]; Available from: <https://www.w3.org/standards/webofservices>.
56. Mascheroni, M.A. and E. Irrazábal, *Framework para la creación y ejecución de pruebas automatizadas sobre servicios REST* 2016: p. 495-504.
57. Buschman, F., K. Henney, and D.C. Schmidt, *Pattern-Orientend Software Architecture*. Vol. Vol. 4. 2007.

58. Larman, C., *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Vol. Vol. 2. 2003.
59. Gamma, E., et al., *Design Patterns: Elements of reusable object oriented programming*. 1995.
60. Guerrero, C.A., J.M. Suárez, and L.E. Gutiérrez, *Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos*. Información Tecnológica, 2013. Vol. 24: p. 103-114.
61. 25000, I. *ISO/IEC 25010*. 2018 07/05/2018]; Available from: <http://iso25000.com/index.php/normas-iso-25000/iso-25010>.
62. Chiu, C.C., *Las pruebas en el desarrollo de software*. 2015, Universidad Nacional Autónoma de México. p. 46.
63. Peño, J.M.S., *Pruebas de Software. Fundamentos y Técnicas*. 2015, Universidad Politécnica de Madrid. p. Pág. 132.
64. Aracil, Y.H. and R.A.C. Prendes, *Sistema basado en casos para la identificación de problemas en los diseños de recursos educativos abiertos*. 2015, Universidad de las Ciencias Informáticas. p. 92.
65. Cuscó, M.B.I., *Talleres para potenciar la motivación en profesores de Educación Física*. Acción, 2017. Vol. 13: p. 37-40.
66. Martí, M.R.V., *Clima docente en estudiantes de la "Misión Sucre" de la Universidad Deportiva del Sur*. Revista Electrónica de Investigación y Docencia (REID), 2014. Vol. 11: p. 165-178.
67. Águila, Y.V., et al., *Sistema Integral de Control Interno para el Vicedecanato de Administración y Servicios de la Facultad 3*. Serie Científica de la Universidad de las Ciencias Informáticas, 2017. Vol. 10: p. 37-51.

Anexos

Anexo No. 1: Encuesta realizada a los diseñadores de recursos educativos

Estimados diseñadores:

Con el empleo de nuevas tecnologías ha aumentado progresivamente la creación de los recursos educativos. Para la confección de estos recursos es necesario el empleo de patrones de diseño, pues estos les proporcionan a los diseñadores las herramientas para construirlos con una sólida estructura.

Actualmente, existe gran cantidad de patrones de diseño de recursos educativos que pudieran utilizarse, estos se encuentran almacenados en diversas fuentes, por lo que se les dificulta la decisión a los diseñadores de recursos educativos de cuál o cuáles aplicar para resolver un problema determinado. Por esta razón, se desarrolló el “Sistema para recomendar patrones de diseño de recursos educativos basado en el tratamiento de agrupaciones y la calidad percibida por los usuarios”.

Con el propósito de que el sistema desarrollado alcance la calidad requerida se hace indispensable recopilar información de los diseñadores de recursos educativos en la universidad. Pedimos su ayuda para que responda las siguientes preguntas. Sus respuestas serán confidenciales y no se reportarán datos individuales. Lea detenidamente cada pregunta y responda sinceramente.

Datos personales del encuestado.

Centro o Departamento docente:	
Categoría docente:	Categoría científica:
Grado científico:	
Años de experiencia utilizando tecnología educativa:	

Preguntas:

1. ¿En su centro o departamento utilizan alguna herramienta de autor para la recomendación de patrones de diseño de recursos educativos?

Sí No sé No

2. Marque con una X la frecuencia con que usted le es necesario la recomendación de patrones de diseño para confeccionar en recurso educativo.

No.	Frecuencia	Selección
1	Muchas veces	
2	Frecuentemente	
3	A veces	
4	Pocas veces	
5	Nunca	

3. ¿Si usted necesitara la recomendación de patrones de diseño para confeccionar un recurso educativo, lo haría mediante el sistema de recomendación propuesto?

Sí No sé No

4. ¿Considera usted que la utilización del tratamiento de agrupaciones de los patrones de diseño contribuye a una mejor recomendación?

Sí No sé No

5. ¿Considera usted que teniendo en cuenta la calidad percibida por el usuario contribuye a una mejor recomendación de patrones de diseño de recursos educativos?

Sí No sé No

6. Marque con una X cuan satisfecho está con el procedimiento propuesto para la recomendación de patrones de diseño de recursos educativos.

No.	Satisfacción	Selección
1	Satisfecho	
2	Más satisfecho que insatisfecho	
3	Me es indiferente	
4	Más insatisfecho que satisfecho	
5	No me satisface	
6	No sé qué decir	

7. ¿Qué otras funcionalidades usted considera que se deban insertar al Sistema de recomendación de patrones de diseño de recursos educativos?

Anexo No. 2: Ejemplo de patrón de diseño

Nombre: ¡Calidad, no cantidad!

Una de las actividades más empleadas por los profesores-tutores de asignaturas o cursos online es la herramienta de comunicación de los SGA denominada foro. Esta herramienta puede utilizarse con diversas finalidades, una de ellas es la intervención de los alumnos con aportaciones propias sobre un tema determinado.

Problema: La intervención no planificada de los alumnos puede generar dos problemas. Por una parte, los alumnos pueden entender que el número de intervenciones será índice de un correcto desarrollo de la actividad, por otra un número elevado de intervenciones por parte de cada alumno en una tutoría numerosa hará muy difícil la lectura de todas las intervenciones y,

mucho más, la respuesta del profesor-tutor. Se puede llegar a situaciones en que el número de intervenciones para una actividad sobrepase el centenar. Se aplicará este patrón, pues, en la planificación de actividades en el foro que requieran la intervención de los alumnos. Su aplicación limitará correctamente el número de intervenciones requeridas, de modo que sea posible una lectura comprensiva por parte de los demás alumnos y las respuestas (evaluadoras, correctoras, etc.) por parte del tutor.

Solución: El profesor-tutor al programar la actividad definirá de forma precisa el número mínimo y máximo de intervenciones en el foro por parte de cada alumno. Subrayará que solo se tendrá en cuenta la calidad de las intervenciones requeridas y que las intervenciones que sobrepasen el número máximo definido no serán tenidas en cuenta o serán penalizadas (en su caso). Se propondrán herramientas alternativas para realizar comentarios que deba conocer el tutor (correo electrónico, chat) quien, en el resumen final de la actividad recogerá las aportaciones significativas realizadas por estos otros medios. La planificación del número de intervenciones permite que, al ser limitado su número, los demás alumnos puedan conocer las intervenciones de cada uno de ellos y el profesor-tutor podrá realizar los comentarios necesarios y adecuados. La planificación del número de intervenciones, en función del tipo de actividad, puede conllevar la pérdida de intervenciones significativas si el tema planteado es muy abierto. Se requiere, pues, un análisis detenido de la adecuación de las intervenciones solicitadas al tipo de actividad propuesto.

Anexo No. 3: Ejemplo de patrón de diseño

Un ejemplo de patrón de diseño para crear una multimedia para enseñar un concepto de programación:

Nombre: Crear un recurso para enseñar un concepto de programación. Contexto: Cuando se aprende un nuevo concepto de programación es valioso poder ver un programa ejemplo, o un fragmento de programa y tener su operación explicada de modo interactivo.

Problema: Para ilustrar adecuadamente el concepto o estructura de programación es útil “correr” el ejemplo. En una clase guiada por un docente esto se logra fácilmente, pero para el auto-estudio un recurso dedicado necesita ser diseñado de modo que permita que el alumno controle el ritmo y progreso a través del proceso de “corrido”.

Solución: Usar una metodología de “multi-frame”: un programa ejemplo, o un fragmento de programa se muestra en un cuadro en donde su ejecución se simula resaltando cada sección del código en el orden de ejecución. En la segunda ventana, el efecto de la ejecución del programa se demuestra a través de una animación, y en la tercera se brinda una explicación de la ejecución. La animación ilustra el código que se está ejecutando resaltando las partes apropiadas del código que se están ejecutando. Al mismo tiempo, la animación también se fracciona y la ventana de texto cambia con cada paso para describir la acción de la línea en particular o sección del código de programa. Así, hay tres partes de la página que transcurren a través de una sincronización: el código, el comentario y la animación. El alumno controla la operación a través de un botón en la ventana de explicación: esto inicia la animación sincronizada.

Anexo No. 4: Ejemplo de lenguaje de patrones de diseño

Nombre: Disposición de la tierra

Los estudiantes reciben una experiencia inicial en el examen de un artefacto, más allá de su capacidad de producción, con la intención de mostrarles la complejidad del campo que están por estudiar.

Problema: A menudo enseñamos cursos que cubren mucho terreno. Si los estudiantes no ven el panorama general bastante temprano, es posible que nunca lo vean mientras se pierden en un mar de detalles. Se gustaría mostrar a los alumnos la amplitud de un tema amplio para que tengan algo con lo que relacionarse y no perderse en los detalles a medida que avanza el curso.

Contexto: Esto tiene una aplicabilidad muy amplia en casi todos los dominios. Es especialmente útil para enseñar temas con muchas partes que deben encajar de ciertas maneras. Como por ejemplo la enseñanza de la programación y la enseñanza de la metodología de diseño. Por lo que se puede usar cuando se haya decidido abandonar los sistemas.

Solución: Muéstrole a los estudiantes un artefacto grande para examinar al principio del curso. Pueden ver de qué se supone en ese curso y qué tipo de cosas se espera que dominen.

El artefacto debe tener la complejidad de algo que le gustaría que pudieran producir al final. Dedique tiempo a examinar las partes y sus interacciones. Puede dedicar más tiempo a las compensaciones inherentes al diseño o no, pero debe mencionar al menos la noción de las concesiones de diseño. El artefacto debe incluir la mayoría de los elementos que son el estudio adecuado para ese curso. Es bueno si el artefacto tiene algunos puntos sutiles. Si las preguntas surgen inicialmente sobre estos puntos, pueden diferirse, pero puede significar que tiene mejores estudiantes de lo que cree. Regrese al artefacto a lo largo del curso, revele y discuta sus puntos sutiles.

Anexo No. 5: Ejemplo de lenguaje de patrones de diseño

Nombre: Error

Se les pide a los estudiantes que creen un artefacto, como un programa o diseño que contenga un error específico. El uso de este patrón explícitamente enseña a los estudiantes cómo reconocer y corregir errores. Se pide al alumno que haga ciertos errores explícitamente y luego examine las consecuencias.

Problema: Los estudiantes a menudo no saben cómo interpretar los mensajes de error provistos por sus herramientas o qué hacer para resolver problemas que son diagnosticados por las mismas. La depuración es una habilidad esencial, ya sea hecha con un sofisticado depurador, o simplemente comparando los resultados reales con las expectativas. Los estudiantes inicialmente no saben qué ocurre cuando se cometen errores y, por lo tanto, no saben qué hacer cuando ven el diagnóstico del programa y resultados incorrectos.

Contexto: Esto es muy aplicable a las primeras etapas de la programación de aprendizaje. La sintaxis y los errores semánticos son frecuentes y los estudiantes deben familiarizarse con los mensajes producidos por los compiladores y los sistemas de tiempo de ejecución y lo que indican sobre el programa.

El lenguaje también podría usarse en un curso de análisis o diseño en el que se requieren ciertos errores específicos, pero bastante comunes, para ser parte del diseño. El instructor puede proporcionar una parte del diseño, que es defectuoso de alguna manera, y se les pide a los estudiantes que completen el diseño sin cambiar la parte del instructor. (Ver también el patrón superior del fijador).

Este lenguaje se usa a menudo en cursos de bases de datos y sistemas operativos donde se les pide a los estudiantes que exploren las condiciones que conducen a un punto muerto, al causarlo específicamente. Del mismo modo, en el aprendizaje de la programación simultánea, a menudo se les pide a los estudiantes que escriban programas en los que las condiciones de carrera entre procesos conducen a la corrupción de los datos.

Solución: Se propone pedir a los alumnos que produzcan un artefacto con ciertos errores específicos (por lo general, un solo error). El efecto del error luego se explora.

Por ejemplo, otorgue a los alumnos una tarea en la que se les indique que creen y ejecuten un programa con ciertos errores específicos. Pídales que comenten sobre los diagnósticos producidos y / o por qué no se produjeron diagnósticos para el error.

Anexo No. 6: Ejemplo de lenguaje de patrones de diseño

Nombre: Rellenar los espacios en blanco

Los estudiantes a menudo pueden aprender un tema complejo al construir varias partes pequeñas de un artefacto más grande. Esto ayuda tanto a sus habilidades de lectura y escritura.

Problema: Los estudiantes principiantes necesitan trabajar en proyectos más grandes de lo que ha sido típico en el pasado, sin embargo, son poco sofisticados y solo tienen un poco de conocimiento y habilidad al principio. Por otro lado, pueden aprender tanto leyendo como haciendo. ¿Cómo puede lograr que los estudiantes trabajen en artefactos más grandes sin abrumarlos?

Contexto: Este lenguaje está destinado a la programación de cursos que intentan mover a los estudiantes rápidamente a material difícil. Probablemente sea posible adaptar esto al diseño también.

Solución: Prepare un programa muy bien diseñado o parte de un programa y elimine algunas partes del código. Muestre el resultado a los estudiantes con instrucciones para completar las partes faltantes. Las partes faltantes deben estar bien especificadas. También es mejor si el resultado se utilizará de inmediato para que los estudiantes puedan ver el efecto de su trabajo.