Universidad de las Ciencias Informáticas

Facultad 4

Título: “Sistema de recomendación de patrones de diseño para Recursos Educativos Abiertos”

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

Autores:

Yosleidy Arteaga Gómez

Yaniel Menéndez Martínez

Tutores:

Ing. Yasirys Terry González

MsC.Ángel Alberto Vazquez Sánchez

Ciudad de la Habana, julio, 2015

“Año 57 de la Revolución”

"Aprende como si fueras a vivir toda la vida y vive como si fueras a morir mañana."

Charles Chaplin

Declaración de Autoría

Declaramos que somos los únicos autores del trabajo titulado:

“Sistema de recomendación de patrones de diseño para Recursos Educativos Abiertos” y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio

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

_____	_____
Yosleidy Arteaga Gómez	**Yaniel Menéndez Martínez**
Autor	**Autor**
_____	_____
Ing. Yasirys Terry González	**MsC. Ángel Alberto Vazquez Sánchez**
Tutor	**Tutor**

DEDICATORIA

A mís padres, abuelo y hermano que son lo más grande que tengo en este mundo.

Yosleidy Arteaga Gómez

A mí madre y mí abuela, que han sabído sacrificarse toda una vída para que hoy yo esté logrando este sueño.

Yaniel Menéndez Martínez

AGRADECIMIENTOS

Llega el momento de agradecer a todas las personas que de una forma u otra formaron parte de este gran sueño que hoy se hace realidad, pues siempre estuvieron presentes durante esta etapa inolvidable de mi vida.

Agradezco a mi madre y a mi padre que son las personas más importantes de mi vida por su entrega y sacrificio, su amor, la confianza depositada en mí, ellos sin dudas, fueron el impulso para conquistar esta meta.

A mi hermano por su apoyo y su cariño.

A mi tía Omayda que ha sido como una madre para mí.

A mi tutora por brindarme su ayuda incondicionalmente cada vez que me hizo falta y por enseñarme cada día un poquito más.

A mi compañero de tesis gracias por su esfuerzo y dedicación en todo este tiempo de investigación y por ser un gran amigo.

A mis profesores por inculcarme buenos valores y transmitirme el conocimiento necesario para que hoy pudiera estar aquí.

A mis amigos pues en ellos encontré una gran familia.

A mis compañeros de aula que me han acompañado en los buenos y en los malos momentos.

Yosleidy Arteaga Gómez

AGRADECIMIENTOS

Quiero agradecerle a mi familia, especialmente a mi abuela y mi madre por haber estado siempre a mi lado apoyándome y por haberme criado de la forma que lo hicieron pues he llegado a donde estoy gracias a ellas.

Agradezco a mi padre que también me ha dado todo su apoyo.

A mi novia que me ha brindado su amor durante los cinco años de mi carrera.

A mis amigos que son todos como una familia para mí y que de una forma u otra han estado siempre a mi lado en los momentos buenos y en los malos.

Agradezco a mi compañera de tesis por haberme ayudado tanto durante toda la carrera.

A mi tutora muchas gracias por enseñarme, aconsejarme y por el tiempo dedicado cada vez que me hizo falta su ayuda.

En fin agradezco a todos los que de una forma u otra influyeron para que yo llegara aquí.

Yaniel Menéndez Martínez

Resumen

Múltiples herramientas han surgido con el objetivo de integrar las Tecnologías de la Información y las Comunicaciones a los procesos de enseñanza - aprendizaje. Esto ha propiciado un incremento de recursos educativos abiertos (REA) que no siempre cumplen con características didácticas - pedagógicas y la calidad requerida para su reutilización, lo que implica un aumento progresivo en la utilización de patrones para el diseño de estos recursos. El uso de patrones ayuda a solucionar problemas con los que frecuentemente se encuentran los diseñadores y permiten ahorrar tiempo y esfuerzo. En ocasiones, se dificulta su localización debido a la heterogeneidad de fuentes que existen, y se torna difícil la decisión a tomar por los diseñadores, de cuál o cuáles patrones aplicar para resolver un problema de diseño de un REA, razón por la que surge esta investigación, que desarrolla un sistema de recomendación (SR) para contribuir a la selección de patrones adecuados para dar solución a un problema de diseño de un REA. Se utilizó 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-gramas contextuales, que permitió comparar las cadenas de textos y recomendar el patrón que más se asemeja al problema que se desea resolver. Se aplicaron las pruebas de caja negra y caja blanca comprobándose el correcto y total funcionamiento del sistema y este fue validado con la técnica de Iadov, evidenciándose que existe una correspondencia satisfactoria entre el objetivo y los resultados obtenidos.

Palabras clave: patrón de diseño, recursos educativos abiertos, sistemas de recomendación

Índice de Contenido

Resumen ..... **VII**

Introducción **1**

Capítulo 1. Fundamentación teórica **5**

1.1. Recursos Educativos Abiertos **5**

1.2. Patrones de diseño **7**

1.3. Sistemas de recomendación **9**

1.3.1. Evolución de los sistemas de recomendación **9**

1.3.2. Clasificación de los sistemas de recomendación **11**

1.4. Sistemas basados en conocimiento **16**

1.5. Análisis de soluciones similares **20**

1.6. Herramientas y tecnologías a utilizar **21**

1.6.1. Metodología de desarrollo de software **21**

1.6.2. Lenguaje de programación **22**

1.6.3. Gestor de base de datos **23**

1.7. Conclusiones parciales ..... ¡Error! Marcador no definido.

Capítulo 2. Concepción del sistema recomendador basado en casos **27**

2.1. CBR Textual **27**

2.1.1. Fase Recuperación de casos **27**

2.1.2. Fase Reutilización de casos **38**

2.2. Patrones utilizados **41**

2.2.1. Patrón arquitectónico **41**

2.2.2. Patrones de diseño **43**

2.3. Integración 44

2.3.1. Conexión 45

2.4. Conclusiones parciales 47

Capítulo 3. Validación del sistema implementado 48

3.1. Pruebas 48

3.1.1. Pruebas de caja negra 50

3.1.2. Pruebas de caja blanca 53

3.2. Técnica de ladov 55

3.3. Conclusiones parciales 59

Conclusiones 60

Recomendaciones 61

Referencias bibliográficas 62

Índice de figuras y tablas

Figura 1. Representación gráfica de los sistemas de recomendación colaborativa 12
Figura 2. Representación gráfica de los sistemas de recomendación basados en contenido 13
Figura 3. Representación gráfica de los sistemas de recomendación basados en conocimiento 14
Figura 4. Representación gráfica de los sistemas de recomendación híbrida 15
Figura 5.Proceso de recomendación 18
Figura 6. Representación gráfica del ciclo CBR (35), (36) 19
Figura 7. Proceso de recuperación de información 28
Figura 8.Integración 47
Figura 9.Prueba de caja negra 50
Figura 10.Clasificación de no conformidades 53
Figura 11.Prueba de caja blanca 53
Figura 12.Nivel de satisfacción 58

Tabla 1. Rasgos 35
Tabla 2. Caso de prueba conexión 50
Tabla 3.Caso de prueba recomendación 51
Tabla 4.Técnica de Iavod 56
Tabla 5.Satisfacción 57
Tabla 6.Resultados 58
Tabla 7.Valores de la fórmula 58

Introducción

En los últimos años se ha propiciado un progreso significativo de las Tecnologías de la Información y las Comunicaciones (TIC). Las TIC juegan un papel fundamental en el desarrollo de todos los sectores de la humanidad. La red de redes es uno de los avances científico-técnico que ha revolucionado el mundo, pues brinda una mayor diversidad de productos y servicios, perfeccionando así los métodos tradicionales de búsqueda de información, lo que implica que disminuya el esfuerzo realizado con los antiguos medios. Esto no significa que la forma tradicional se deseche, sino que haya una mayor variedad para la selección de acuerdo a la necesidad de los usuarios.

La World Wide Web (www) como la principal fuente de información y de relaciones interpersonales, se ha convertido en una apuesta segura para el presente y el futuro de la educación. La sociedad actual basa su proceso educativo sobre una perspectiva constructivista pues lo más importante no es el cúmulo de información adquirido, sino la calidad de esta y la capacidad de poder utilizarla en situaciones concretas y adaptarla al medio. Internet ha creado un aula gigante en la que millones de usuarios están conectados y relacionados compartiendo y enriqueciendo cada vez más sus conocimientos.(1)

Con la introducción de las TIC en esta esfera, surge la educación a distancia que ha abierto un nuevo camino en las formas de adquisición del conocimiento, enriqueciendo la modalidad presencial con nuevas formas virtuales.En la educación a distancia se utilizan REA para mejorar el proceso de enseñanza y aprendizaje.

El término REA ha sido utilizado para referirse a los materiales con fines educativos, tales como: objetos de aprendizaje, audio-conferencias, audio-video conferencias, imágenes y sonidos, el contenido de un curso, colecciones de artículos de revistas, libros de texto, que pueden ser adquiridos libremente.(2),(3)

En Cuba, el auge alcanzado por la creación de REA ha conllevado a la utilización de estos recursos en todos los niveles de enseñanza. La Universidad de las Ciencias Informáticas (UCI), que vincula la docencia-producción-investigación como modelo de formación profesional, cuenta con varias facultades. En ellas, se han creado varios REA para apoyar la educación en Cuba y en otros países, como Venezuela. Especial implicación han tenido los proyectos CRODA y RHODA donde se desarrollan herramientas para producir y almacenar REA. En algunos casos se han presentado problemas con estos recursos en cuanto a diseño y esto ha traído consigo la necesidad de arreglar defectos en ellos.

En muchas ocasiones, se han realizado soluciones similares que previenen o resuelven el mismo problema en los diseños de REA, generando de esta manera trabajos paralelos, lo que provoca empleo innecesario de tiempo y de recursos. Para resolver esta situación se utilizan los patrones de diseño en la producción de los recursos educativos abiertos.

Los patrones en sentido general son guías, directrices, plantillas que se generan cuando se resuelven muchas problemáticas que tienen grandes semejanzas y, por consecuente, se crea un modelo de respuestas que se puede reutilizar para no realizar trabajos repetidos en el análisis de una posible solución del problema que ya ha sido resuelto. Al patrón se le añaden cambios oportunos o pequeñas particularidades de la situación en la que se aplique.(4)(5)(6)

Los patrones de diseño de REA están almacenados en diferentes repositorios, que no necesariamente son exclusivos para este tipo de patrones. La situación antes descrita, implica que se les dificulte a los diseñadores la búsqueda de estos patrones, que comúnmente se convierte en un proceso extenso.

Además, ha ocurrido que aunque se tenga identificado el problema en el diseño de REA muchas veces los diseñadores, fundamentalmente los menos experimentados, tienen dificultad para determinar, cuándo reutilizar un patrón de diseño o no están seguros de cuál sea la solución más adecuada, provocando la creación de patrones nuevos, que en algunos casos son similares a los que ya existen.

De la situación antes expuesta se origina el siguiente **problema de investigación** ¿Cómo contribuir a la selección de patrones adecuados para solucionar problemas de diseño de un recurso educativo abierto?

Se define como **objeto de estudio**: el proceso de recomendación de patrones, siendo el **campo de acción**: el proceso de recomendación de patrones de diseño para REA.

El **objetivo general** de la investigación es: desarrollar un sistema de recomendación que contribuya a la selección de patrones adecuados para solucionar problemas de diseño de un REA.

Para los efectos de esta investigación, se define un sistema de recomendación como un programa informático que sugiere al usuario una decisión a tomar ante un problema, a partir de las preferencias del usuario, o de conocimiento almacenado en una base de datos o a partir de la unión de estos dos.(7), (8)

Del objetivo general se derivan los siguientes **objetivos específicos**:

- Fundamentar los conceptos, características y antecedentes de los sistemas de recomendación mediante un estudio bibliográfico.
- Seleccionar las técnicas de recomendación que se van a utilizar.
- Implementar el sistema de recomendación de patrones de diseño para recursos educativos abiertos.
- Validar el sistema implementado.

Para dar cumplimiento a los objetivos propuestos se tiene como **hipótesis** que si se desarrolla un sistema de recomendación se contribuirá a la selección de patrones adecuados para solucionar problemas de diseño de un REA.

Métodos teóricos:

Para la realización de esta investigación se utilizarán métodos que permitirán reproducir teóricamente el objeto, en el pensamiento, estudiar las características del objeto de investigación que no son observables directamente y facilitarán la construcción de ideas de investigación. Los métodos que serán empleados son:

Histórico– Lógico: este método será usado para hacer un estudio acerca del origen de los SR 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 estos sistemas y sirva esto como base para el desarrollo del sistema a desarrollar 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, sobre los patrones de diseño de REA y sobre las técnicas de búsquedas y recuperación de información, 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.

Estructuración del contenido con una breve explicación de sus partes. El presente trabajo consta de una introducción, tres capítulos, conclusiones generales, recomendaciones, referencias bibliográficas

utilizadas y, por último, los anexos que complementan el cuerpo del trabajo. A continuación se explicará de forma breve en que consiste cada capítulo.

Capítulo 1: Fundamentación teórica

Se definen los conceptos de REA y de patrones de diseño, se describen las principales características de los SR, cómo han evolucionado desde su surgimiento y se realiza la selección de las tecnologías y herramientas actuales, que se van a utilizar para el desarrollo del sistema.

Capítulo 2: Concepción del sistema recomendador basado en casos

Se describen las etapas del ciclo del razonamiento basado en casos que serán utilizadas para la implementación del sistema, explicando las técnicas y los algoritmos de recuperación de casos textuales que se utilizarán en la recomendación. Además, se describen los patrones arquitectónicos y de diseño que serán utilizados en la implementación de la aplicación.

Capítulo 3: Validación del sistema implementado

Se realizan diferentes pruebas al sistema obtenido como resultado de la investigación, utilizando técnicas que se enfocan en verificar la efectividad de las aplicaciones afines a lo que se desea.

Capítulo 1. Fundamentación teórica

Introducción

En el presente capítulo se describen los conceptos y elementos teóricos que permiten comprender las características principales de los REA y de los patrones de diseño de estos recursos. También, se da una panorámica de la evolución de los SR desde sus inicios hasta la actualidad y se estudian los tipos de SR que existen. Además, se realiza una breve explicación sobre las principales tecnologías y herramientas que se utilizarán en la solución propuesta.

1.1. Recursos Educativos Abiertos

El sistema de recomendación que se va a desarrollar es para la sugerencia de patrones de diseño que permitan resolver problemas de los REA. El término de REA es muy amplio y existen varios conceptos que definen su significado.

La UNESCO, en el 2002 se refiere a los REA como la provisión abierta de recursos educacionales mediados por las tecnologías de la información y las comunicaciones para consultas, uso y adaptación por una comunidad de usuarios con propósitos no comerciales.(9)

Ya en el 2007, la Fundación William y Flora Hewlett, promovió una definición más específica sobre los recursos educativos abiertos, pues incluye varios ejemplos de estos recursos, al plantear que: "Los REA son recursos para la enseñanza, el aprendizaje y la investigación, que residen en el dominio público o han sido publicados bajo una licencia de propiedad intelectual que permite que su uso sea gratuito. Los recursos educativos abiertos incluyen: cursos completos, materiales para el curso, módulos, libros de texto, videos, pruebas, software y cualquier otra herramienta, materiales o técnicas utilizadas para apoyar el acceso al conocimiento.(10)

Otro concepto asociado al término "recurso educativo abierto" es definido por la Organización para la Cooperación y el Desarrollo Económico (OCDE) como materiales digitales ofrecidos gratis y abiertamente a profesores, estudiantes y aprendices autónomos para ser usados y reutilizados en la enseñanza, el aprendizaje y la investigación.(11)En esta definición ya se incluyen los usuarios a los que van destinados estos recursos y que, por tanto, se van a favorecer con ellos.

A partir de estas definiciones y otras estudiadas (12),(13),(14)las características que se destacan de los REA son:

- Su objetivo es apoyar el proceso de enseñanza y aprendizaje a partir de su propósito educativo.
- Pueden ser utilizados por todas las personas de forma abierta y gratis.
- 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.
- Su distribución no es restringida.

Según la Fundación para el Conocimiento Abierto, el conocimiento implícito en los REA debe cumplir las siguientes condiciones para que sea abierto(15),(16),(17):

- Accesibilidad: el conocimiento debe estar disponible integralmente y en un formato conveniente para ser modificable. El costo de adquisición debe ser razonable, y en el mejor de los casos que se pueda descargar de manera gratuita en Internet.
- Redistribución: la distribución de la obra no debe ser restringida por una licencia, además de que no se debe exigir un pago u otro tipo de cuota mediante la licencia.
- Reconocimiento: mediante la licencia se puede exigir que para la redistribución y la reutilización, se reconozcan los creadores de la obra.
- Integridad: la licencia puede requerir que cuando se cree una modificación sobre la obra el trabajo derivado tenga un nombre o número de versión diferente al original.
- Reutilización: en la licencia se debe permitir que se realicen modificaciones a la obra original, y que se puedan distribuir las derivaciones sin ninguna discriminación y sobre las mismas condiciones que la original, aunque se puede imponer algún tipo de requerimiento referente al reconocimiento y a la integridad.
- Ausencia de restricciones tecnológicas: no debe haber ningún obstáculo tecnológico para ejecutar los actos mencionados anteriormente cuando se proporcione la obra.
- Sin discriminación de personas o grupos.

- Sin discriminación de ámbitos de trabajo.

1.2. Patrones de diseño

La definición que crea el pilar inicial de este tema es la dada por Christopher Alexander en 1977, donde propuso la idea del patrón como aquel que: “describe un problema que ocurre una y otra vez y, a continuación, describe el núcleo de la solución de ese problema, de tal manera que el usuario puede utilizar esta solución un millón de veces más, sin tener que hacerlo de la misma manera dos veces”. (18)

Miguel Zapata aporta otro concepto más específico donde pone de manifiesto algunos contextos donde se puede utilizar un patrón. Él plantea que los patrones son: “Estructuras de información que permiten resumir y comunicar la experiencia acumulada y la resolución de problemas, tanto en la práctica como en el diseño, en programas de enseñanza y aprendizaje a través de redes.” (19)

Cuando se trata de patrón de diseño de un REA se puede decir que:

Se entiende como patrón de diseño la estructura pedagógica que tienen los recursos de aprendizaje en su construcción, capaz de articular diversos elementos informativos, y que se constituyen en un componente constante, altamente configurable y actualizable.(20)

Se destacan dos características fundamentales en los patrones: reusabilidad y flexibilidad.

Cuando se diseña un patrón habrá que atender a la idea de realizar un trabajo reusable pues como lo dice el propio concepto se puede utilizar esta solución un millón de veces más y debe permitir su aplicación en diferentes contextos. De otra forma, se limitaría a la solución de problemas muy concretos, con lo que seguiría en la misma dinámica de inversión de esfuerzo continuo sin aprovechar los conocimientos y la experiencia acumulada.(21)

Vistas ya estas características de los patrones de diseño se procede a explicar los elementos básicos que debe tener un patrón, aunque se pueden incluir otros más(19):

- Nombre

Cada patrón debe identificarse con un nombre que además de ser descriptivo del problema-solución que representa, lo ayude a relacionar con otros patrones diferentes.

- Contexto

Describe las condiciones en las que se desarrolla el problema. Debe ser lo suficientemente general para ser aplicadas en casos muy diferentes dentro de un contexto, pero aún lo suficientemente específico como para dar orientaciones constructivas.

La descripción del contexto, ayuda a su aplicación futura y a la construcción de nuevos patrones derivados. Es una parte indispensable en un patrón de diseño puesto que se podrá conocer el marco de aplicación para el patrón, así como marcarán en gran medida sus condiciones de flexibilidad y reusabilidad.

- Sistema de fuerzas

Se refiere al objetivo o problema que el patrón afronta. Los patrones de aprendizaje nacen de la confrontación de dos posturas en tensión que genera un conflicto a resolver.

- Solución

Es la configuración del sistema, de las condiciones disponibles a partir del contexto anteriormente descrito, para lograr un equilibrio entre las dos tensiones contrapuestas que han creado el problema y, por tanto, la necesidad de un patrón.

Para comprender mejor todos estos elementos que componen un patrón se muestra un ejemplo de patrón de diseño de un REA(22):

Nombre del patrón: Creación GLO (objeto de aprendizaje generacional)

Clasificación del patrón: Creacional

Contexto: Repositorio de RLO (CETL) (repositorio de objetos de aprendizaje)

Sistema de fuerzas o problema: Crear objetos de aprendizaje personalizados y adaptados para un proceso de enseñanza aprendizaje específico.

Solución: Separar del RLO (objeto de aprendizaje reusable) el contenido del diseño, de esta manera se obtendrán objetos más complejos y potentes que partiendo de un diseño genérico pueden ser personalizados para su uso generalizado en distintas ámbitos.

Para ver más ejemplos de patrones ir al anexo 1.

La importancia del uso de patrones o plantillas para crear cualquier tipo de elemento ha quedado demostrada durante el transcurso de la humanidad, desde los patrones usados en la industria textil para el diseño y la creación de prendas de vestir hasta los patrones de diseño en ingeniería de software. Sin embargo, los patrones de diseño para REA son más que plantillas, pues aportan al proceso de creación de los mismos una organización interna de los objetos que lo componen.

De todo lo antes planteado, se puede concluir que los patrones de diseño de REA pretenden evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos, creando un estándar en el diseño de soluciones con la formalización de un lenguaje común a todos los diseñadores. Por ende, se facilita el aprendizaje a las nuevas generaciones que van condensando el conocimiento existente y disponen de un método ya conocido para la creación de nuevos patrones. No pretendiendo en ningún caso coartar la libertad creativa de cada persona que puede aplicar el patrón de la manera que más le convenga o guste, así como puede ofrecer nuevas alternativas. (21)

1.3. Sistemas de recomendación

1.3.1. Evolución de los sistemas de recomendación

Debido al gran cúmulo de información que existe en la actualidad, se dificulta mucho el proceso de búsqueda, reduciendo las capacidades para procesarla información por parte de los usuarios. Por tal motivo, es el momento de desarrollar tecnologías que alivien esta sobrecarga de información y surgen los sistemas de recomendación, como una alternativa.

Los sistemas de recomendación se encargan de proporcionar a los usuarios consejos e información sobre productos y/o servicios que puedan ser de interés a la hora de tomar una decisión. Este proceso, en el que el sistema guía al usuario a la hora de realizar una elección, puede proporcionar resultados que sean de gran utilidad, ya sea ahorrando tiempo, proporcionando datos relevantes de forma precisa y fácil, e incluso obteniendo información que permite valorar opciones que de otra forma antes no se habrían contemplado, algo muy apreciado por la mayoría de los usuarios. (23)

Otro concepto asociado a los SR es: Los SR son programas informáticos que estudian los intereses y preferencias delos consumidores individuales [...] y hacen recomendaciones al respecto. Tienen el potencial para apoyar y mejorar la calidad de las decisiones que toman los consumidores durante la búsqueda y selección de productos en línea.(24)

Ricci en el 2011 planteó que los SR son herramientas de software y técnicas que proporcionan sugerencias para que los artículos sean de utilidad para el usuario. Las sugerencias se refieren a diversos procesos de toma de decisiones, como qué artículos para comprar, qué música escuchan, o qué noticiasen línea para leer.(7)

De estas definiciones se puede concluir que los SR son programas inteligentes que sugieren al usuario una opción adecuada ante una decisión a tomar.

Los SR han evolucionado y es posible encontrarlos en diversos ámbitos de aplicación, donde se han convertido en una herramienta fundamental para los proveedores y los servicios de información.

Tapestry fue el primer sistema de este tipo surgido en la década de los 90, que permitía almacenar el *feedback* de los usuarios sobre los artículos o noticias que estos habían leído, para que, posteriormente, esto fuera utilizado por otros usuarios que aún no habían leído el artículo o noticia, permitiendo establecer si la información del documento era relevante o no.(25)

Otra de las primeras formas de filtrado de información electrónica apareció con el trabajo de Housman y Kaskela, quienes diseñaron un método que de forma automática mantenía a los científicos informados sobre nuevas publicaciones en sus áreas de trabajo o especialización. Este sistema creaba un perfil de usuario el cual contenía ciertas palabras claves relevantes para el usuario, que luego las utilizaba para buscar coincidencias entre estas y los nuevos documentos o artículos publicados, con el fin de recomendar cuáles de las informaciones encontradas serían del interés para los científicos.(26)(27)

The information Lens System fue una aproximación a los sistemas recomendadores en donde, basado en el contexto del correo electrónico, se les permitía a los usuarios crear reglas para filtrar los correos electrónicos. Así pues, los usuarios podían, por ejemplo, crear una regla para etiquetar todos los mensajes provenientes de cierta persona o correo electrónico. Gracias a este sistema, las personas podían ordenar y priorizar los mensajes que se dirigían a ellos, y también les ayudaba a encontrar mensajes útiles que no habrían recibido. El problema de *The information Lens System* apareció con las personas o usuarios cuyo conocimiento en el ámbito de la informática era elemental o casi nulo. Estos usuarios eran incapaces de crear reglas de filtrado para priorizar o filtrar los correos recibidos.(26),(28),(29)

En 1997 es que surge el término de sistema de recomendación, introducido por Resnick y Varian basándose en dos razones: en primer lugar porque puede ocurrir que los usuarios no colaboren

explícitamente entre ellos y en segundo lugar, porque el sistema puede sugerir elementos no conocidos por el usuario hasta el momento y en ese caso se estaría realizando una recomendación.(30)

1.3.2. Clasificación de los sistemas de recomendación

El objetivo de los sistemas de recomendación es guiar al usuario a elegir una meta que satisfaga sus necesidades, pero para cumplir con sus expectativas no todos los sistemas utilizan las mismas técnicas, difieren unas de otras, tanto en la información requerida como en los procesos necesarios para llevar a cabo las recomendaciones.

Estos sistemas se pueden clasificar en(7)**,** (31)**,** (32)**:**

- Recomendación colaborativa
- Recomendación basada en contenido
- Recomendación basada en conocimiento
- Recomendación híbrida

Recomendación colaborativa

Se les llama también sistemas de recomendación sociales. El filtrado colaborativo a la hora de ofrecer una recomendación a un usuario no considera únicamente las preferencias personales, sino también las de otros usuarios con intereses similares a los suyos. Este tipo de recomendación consiste en sugerirle a un usuario que pertenezca a un grupo, los ítems que les hayan gustado a usuarios que pertenezcan al mismo grupo y que tengan semejantes gustos y preferencias. Este tipo de filtrado es muy efectivo para cuando se dispone de bases de datos amplias.(7), (31)**,** (32)

El filtrado colaborativo es considerado una de las tecnologías más potentes de personalización dentro del amplio concepto de la web adaptativa. Una de las aplicaciones basada en este tipo de sistema es amazon.com que es uno de los principales sitios de ventas por la web. Las limitaciones fundamentales del filtrado social son(7), (31)**,** (32):

- No es posible hacer recomendaciones hasta que el elemento a analizar no esté bien configurado o esté lo suficientemente completo para definir su grupo de vecinos cercanos.

- Este tipo de sistemas tiende a ofrecer resultados pobres cuando se dispone de poca información de los elementos a analizar o las preferencias a considerar son muy heterogéneas.

Para comprender mejor en qué consiste el filtrado colaborativo se muestra la figura 1.

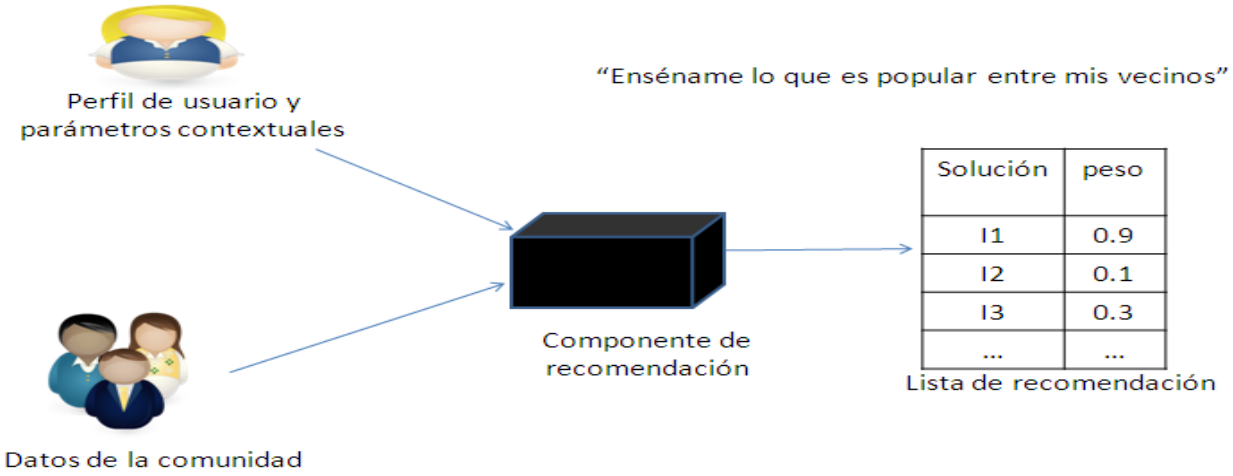


Figura 1. Representación gráfica de los sistemas de recomendación colaborativa(25),(31),(32)

Recomendación basada en contenido

Este tipo de sistema aprende del perfil del usuario, basándose en la jerarquía que el usuario creó en su perfil respecto a las características de los ítems de este. Extrae características de los elementos no conocidos aún por el usuario actual y las comparan con el perfil del mismo para predecir sus preferencias sobre estos. Si las características entre ambos ítems son semejantes, entonces el sistema genera la recomendación al usuario. La ventaja de las recomendaciones basadas en contenido es que se pueden realizar recomendaciones sin la necesidad de tener un historial previo, por lo que permite realizar predicciones independientemente del historial del usuario, aunque estos también presentan una gran dificultad, la sobre-especialización, debido a que no tiene en cuenta los posibles cambios en los gustos de los usuarios.(7), (31)**,** (32)

Este tipo de sistemas suele ser utilizado en recomendación de páginas web, artículos noticiosos, restaurantes, programas de televisión, y artículos para vender. Un ejemplo de esto es el sistema de recomendación *Entree*, un sistema que recomienda restaurantes en la ciudad de Chicago. *Entree* pertenece a la familia de los sistemas *FindMe* que fueron de los primeros recomendadores que

consiguieron incluir retroalimentación por parte del usuario en forma de críticas a las recomendaciones propuestas.

Los pasos a seguir en las recomendaciones basadas en contenido son los siguientes(7), (31)**,** (32):

- Extraer las características de los objetos.
- Comparar los objetos con el perfil del usuario para identificar las preferencias de este sobre dicho objeto.
- Recomendar objetos similares en su contenido a los que forman parte de su perfil.

Para ver el proceso antes descrito se muestra la figura 2.

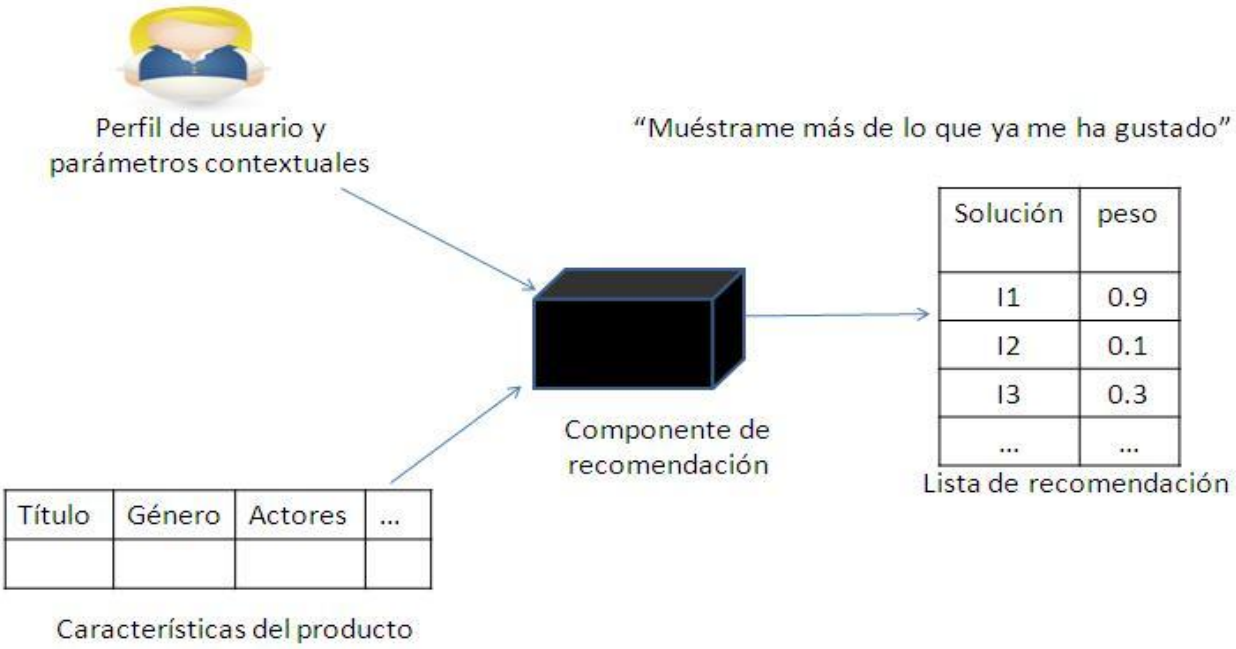


Figura 2. Representación gráfica de los sistemas de recomendación basados en contenido(31),(32)

Recomendación basada en conocimiento

Dispone de información acerca de cómo un ítem satisface una necesidad del usuario y de acuerdo a esta necesidad genera la recomendación. Algunos sistemas han usado el razonamiento basado en casos para la recomendación. Los sistemas basados en conocimiento no dependen de grandes cantidades de información sobre objetos puntuados y usuarios particulares, sino que lo único que necesitan es tener un conocimiento general sobre el conjunto de elementos y las necesidades del usuario. (7), (31)**,** (32)

Un ejemplo de estos tipos de sistemas que basan su funcionamiento en el conocimiento son los buscadores como Google, que funciona de la siguiente forma: primero el usuario realiza la petición, luego el sistema comienza a generar un rastreo e indexación que básicamente es seguir enlaces de una página a otra que tengan coincidencias con la petición realizada y después las va ordenando según el contenido para almacenarlas en un índice. El paso que sigue es la utilización de algoritmos para recomendar de forma rápida y eficiente la información que más se ajuste a la petición realizada, en general, todos los buscadores trabajan de forma similar lo único que los puede diferenciar son la calidad de los algoritmos que utilicen para hacer las búsquedas y recomendaciones.

El principal problema de estos sistemas es que requieren un gran esfuerzo humano para realizar las recomendaciones, además utilizan una base de datos considerablemente grande y bien seleccionada. La mayoría de las veces el usuario lo que busca no es un producto exactamente igual al producto ejemplo sino 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.(7), (31)**,** (32)

Para comprender mejor en qué consisten los sistemas recomendadores basados en conocimiento se muestra la figura 3.

Figura 3. Representación gráfica de los sistemas de recomendación basados en conocimiento(33),(34),(35)

Recomendación híbrida

Todos los sistemas de recomendación anteriormente expuestos tienen sus ventajas y sus desventajas por lo que se hace necesario en ocasiones combinarlos entre sí para obtener un mejor resultado. La principal desventaja dela recomendación híbrida es que su implementación añade complejidad al sistema y requiere, además, la construcción de dos o más sistemas de recomendación así como la necesidad de interoperabilidad entre ellos.(7), (31)**,** (32)

En la siguiente figura se muestra el proceso de recomendación híbrida.

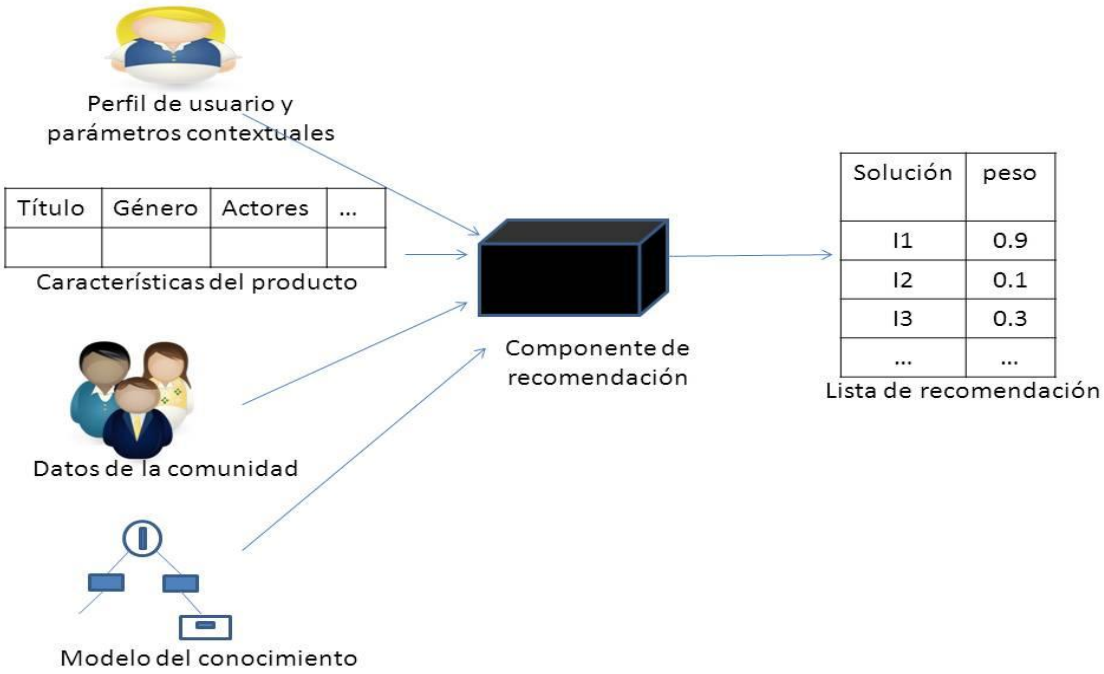


Figura 4. Representación gráfica de los sistemas de recomendación híbrida(31),(32)

Sistema de recomendación que se utilizará

De los tipos de sistemas de recomendación analizados el que se aplicará para resolver el problema de investigación, es el basado en conocimiento, pues utiliza el conocimiento sobre un dominio para arribar a la solución de un problema en ese dominio. En el caso del sistema que se desarrolla el conocimiento del que se dispone es sobre los patrones de diseño de REA para dar solución a problemas de REA. Para esto no es necesario disponer de un grupo de usuarios para recomendar según el grupo al que pertenezca el

diseñador, ni es importante conocer sus preferencias, sino tener una base de conocimiento que permita resolver una necesidad al diseñador, sin tener en cuenta sus características particulares.

1.4. Sistemas basados en conocimiento

Los sistemas basados en conocimiento pretenden representar funciones cognitivas del ser humano como el aprendizaje y el razonamiento. Esta clase de aplicaciones descansan en las contribuciones de la Inteligencia Artificial. Su orientación es la automatización del análisis de problemas, la búsqueda de soluciones, la toma de decisiones y el empleo de conocimiento especializado en un campo específico de aplicación.(33), (34), (35)

La composición de los sistemas basados en conocimiento consta de: un mecanismo de aprendizaje, una base de conocimiento, un motor de razonamiento, y medios de comunicación hombre-máquina.(33),(34), (35)

Entre sus propósitos destacan(33):

- Aprender
- Evolucionar
- Adaptar
- Razonar
- Tomar decisiones
- Analizar problemas
- Generar alternativas de solución
- Generar conocimiento a partir del que ya se posee

El sistema de recomendación que se desarrolla como resultado de la presente investigación tiene dentro de sus propósitos razonar, analizar problemas, tomar decisiones y generar alternativas de solución.

Entre las características más relevantes de los sistemas basados en conocimiento están(33):

- Maneja la incertidumbre.
- Genera 0, 1 ó múltiples soluciones.

- Provee conocimiento técnico y científico.
- Busca generar la solución “óptima”.
- Aprende de los fracasos.
- Emplea métodos para la representación del conocimiento.
- Posee capacidad para realizar su propio razonamiento, cuestionamiento y emisión de conclusiones.
- Utiliza búsquedas heurísticas.
- Pueden utilizar razonamiento con base en probabilidades, creencias, pertenencias y suposiciones.

De estas características las que poseerá el sistema de recomendación que se va a desarrollar son: generar 0,1 o múltiples soluciones pues si no encuentra en la base de conocimiento un problema semejante al que entró el usuario no devuelve ningún patrón, sin embargo, si encuentra un problema o más semejantes al entrado por parámetro el sistema recomendará el(los) patrón(es) asociado(s) a ese problema. Otra característica que tendrá en cuenta el sistema recomendador de patrones será el empleo de métodos para la representación del conocimiento.Además, el sistema recomendador tendrá la característica de poseer capacidad para realizar su propio razonamiento, cuestionamiento y emisión de conclusiones a través de algoritmos y métodos de recuperación de información y emitirá una conclusión que será el patrón recomendado para resolver el problema de diseño de REA. Otra característica que se tendrá en cuenta para el desarrollo del sistema es la utilización de razonamiento con base en probabilidades pues se recomendará el patrón que más probabilidad tenga de resolver el problema de diseño de REA.

Tipos de sistemas basados en conocimiento:

Sistemas de razonamiento basado en casos (CBR, del inglés *Case Based Reasoning*)(34)**,** (35):

- Utilizan una función de similitud que estima cuánto de lo que el usuario necesita (descripción del problema) coincide con los casos almacenados. El valor que toma esta función se interpreta directamente como la utilidad que tiene la recomendación para el usuario.(34), (35)
- Facilitan la adquisición de conocimiento debido que solo hace falta un conjunto de problemas resueltos que sirva de base de conocimiento (base de casos).

- El CBR es el modo natural de razonar de los seres humanos. Además, permite el aprendizaje del propio sistema ya que, el sistema añade el problema resuelto como un nuevo caso para la base de casos.(34), (35)

Sistemas de razonamiento basado en restricciones (RBR, del inglés *Restrictions Based Reasoning*)(34)**,** (35):

- Con el razonamiento basado en reglas se construye el motor de inferencia mediante reglas de la forma *"*Si A entonces B*",* estas reglas se disparan cuando se cumplen sus antecedentes A y provocan cambios con sus consecuentes B en la base de conocimiento.(34),(35)

- Similar a los basados en casos desde el punto de vista del conocimiento utilizado, la diferencia entre ellos es que los basados en casos usan métricas de similitud, en tanto estos utilizan por lo general bases de conocimiento que contienen reglas explícitas que indican cómo relacionar los requerimientos del usuario con las características del tópico.(34),(35)

Ambos realizan el mismo proceso de recomendación tal como se muestra en la figura 5:

Figura 5.Proceso de recomendación

Tipo de sistema basado en conocimiento que se utilizará

Después de haber analizado los dos tipos de sistemas recomendadores basados en conocimiento, se decide utilizar en el sistema a desarrollar el basado en casos, pues se apoyará la recomendación

principalmente en los problemas de diseño de REA que ya han sido resueltos anteriormente y estarán almacenados en el modelo de conocimiento, buscando semejanzas entre estos y el problema actual a partir de métricas de similitud.

Hay dos tipos básicos de sistemas basados en casos: de interpretación de situaciones o de resolución de problemas. Los primeros se basan en formular un juicio o clasificar una situación, por ejemplo diagnosticar una enfermedad según unos síntomas. En este tipo de sistemas los casos son fáciles de representar y recuperar, y la adaptación es muy sencilla o nula. Los sistemas CBR de resolución de problemas se basan en aplicar la solución de un problema anterior para obtener la solución al problema actual. (35)

El sistema recomendador que se va a implementar será el basado en casos resolvedor de problemas, pues a partir de la base de casos que contiene los patrones de diseño se obtendrá la solución del problema actual.

El razonamiento basado en casos se explica mediante el ciclo CBR que tiene cuatro fases(34),(35):

- **Recuperar** los casos más relevantes para una consulta dada.
- **Reutilizar** estos casos para obtener una primera solución.
- **Revisar** la solución construida, adaptándola a la situación actual.
- **Recordar** la solución final para añadirla a la base de casos.

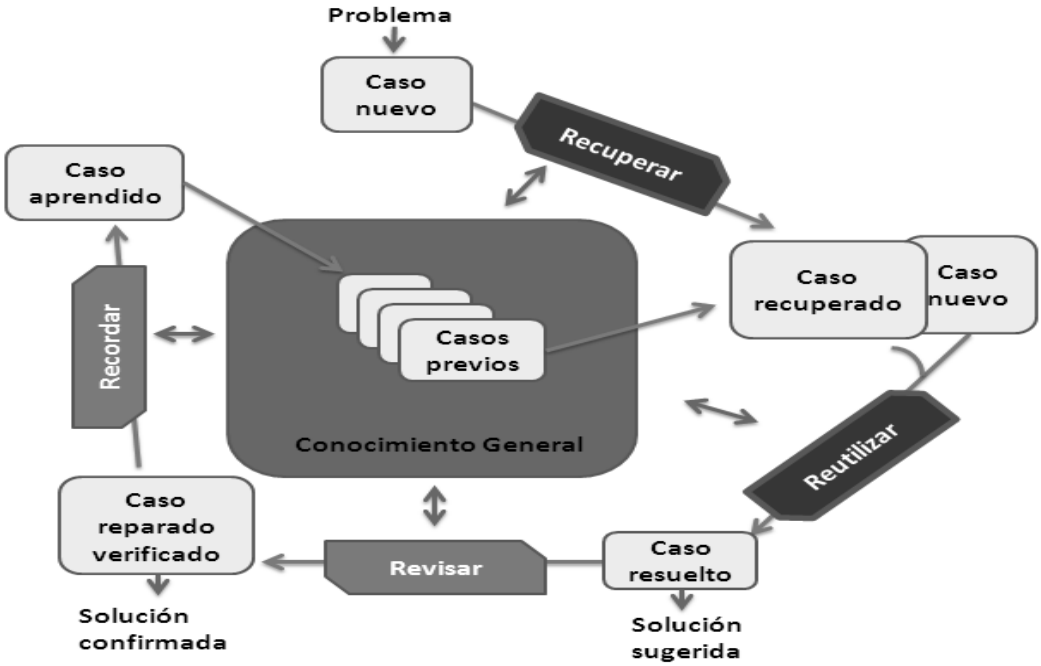


Figura 6. Representación gráfica del ciclo CBR (35), (36)

Este ciclo no es necesario realizarlo completo, se realizan las fases que permitan solucionar el problema de investigación que presente el desarrollador de software.(35), (36)

El sistema que se desarrolla solo utiliza para la implementación dos de estas fases, que serían la recuperación de casos y la reutilización, pues en la fase de reutilización se genera una solución y esta será la sugerida por el sistema de recomendación, ya así, se resuelve el problema planteado en la investigación, pues, se recomienda un patrón de diseño de REA, a partir de patrones almacenados en la base de casos para dar solución a un problema de REA. Por tanto la fase de revisar y recordar no serán utilizadas en el desarrollo de la aplicación, ya que además, la solución sugerida se modifica y se almacena haciendo uso de otras herramientas desarrolladas para este fin. Para la recomendación, después de recuperar varios patrones que resuelven un problema semejante al entrado por el diseñador de REA, se selecciona el que más se ajusta y se sugiere como solución.

1.5. Análisis de soluciones similares

Se realizó un detallado estudio a tres sistemas de recomendación desarrollados en la Universidad, el primero se titula “Implementación de algoritmos de recomendaciones de imágenes para textos noticiosos”(37), en el cual se utiliza la recomendación basada en conocimiento para sugerir imágenes al usuario. Estas imágenes son utilizadas para representar una noticia determinada. Este sistema no es posible utilizarlo para resolver el problema que se plantea en esta investigación, porque las imágenes son recomendadas a partir de la comparación entre metadatos que son identificados de ellas y palabras claves extraídas del contenido del texto de la noticia. Pero el sistema que se va a desarrollar como resultado de la investigación, realiza la recomendación a partir de la comparación entre el problema que resuelve el patrón almacenado en la base de casos y la necesidad del diseñador de REA. En la base de casos no hay metadatos de los patrones, sino el patrón en sí.

El segundo sistema analizado fue el que se titula “Sistema de recomendación basado en las preferencias de los usuarios para la plataforma PTARTV” (Plataforma de Trasmisión Abierta para Radio y Televisión) (38), en el cual se utiliza la recomendación basada en contenido. Este no resuelve el problema planteado en la presente investigación pues se basa en el perfil de los usuarios para recomendar programas de radio y televisión dependiendo de sus preferencias, y en el caso del sistema que se va a desarrollar no se necesita conocer datos de los usuarios para realizar la recomendación.

Se estudió, además, el sistema titulado “Concepción y desarrollo de un sistema de recomendación para jurados online de programación”,(39)enfocado a orientar a los estudiantes con respecto a qué ejercicios resolver dependiendo de sus características utilizando la recomendación basada en contenido, por lo tanto, no puede ser utilizado para el desarrollo del sistema que se va a implementar porque este no necesita de un perfil de usuario para recomendar.

Por último, se realizó un análisis del sistema desarrollado en la Universidad Complutense de Madrid titulado “Catalogador automático de textos y recomendador de artículos del Portal de Revistas Electrónicas de la biblioteca UCM”(35), que realiza la recomendación híbrida combinando el filtro colaborativo con el basado en conocimiento. La biblioteca de la UCM(35) cuenta con conocimiento en diferentes formas: libros, artículos, tesis, bases de datos referenciales. Este sistema realiza búsquedas que tienen que ver con el término o los términos que introdujo el usuario, o búsquedas que muestran como resultado una recomendación de libros o documentos porque otros usuarios similares los consultaron previamente. Este sistema no es factible utilizarlo porque utiliza la recomendación híbrida, donde necesita de una base de conocimiento y además de los perfiles de usuario para generar la recomendación y el sistema que se desarrollará no trabaja con los perfiles de usuario.

Después del análisis realizado a varios sistemas se llega a la conclusión que es importante hacer un nuevo sistema de recomendación basado en casos para resolver el problema planteado en la investigación.

A continuación, se describen las herramientas y tecnologías que se utilizarán para el desarrollo del sistema de recomendación de patrones de diseño de REA.

1.6. Herramientas y tecnologías a utilizar

1.6.1. Metodología de desarrollo de software

Una metodología es un conjunto de filosofías, etapas, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación que permiten guiar a los desarrolladores de aplicaciones.(40)

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería.(40)

Fundamentación para la no utilización de metodología de desarrollo de software

Se puede plantear que para el desarrollo del SBC no se hace necesario el uso de metodologías ágiles o robustas de desarrollo de software, fundamentalmente porque como se explicó en el epígrafe 1.4.1, los SBC plantean su propio ciclo de vida para el desarrollo de aplicaciones, donde cada etapa identifica los procedimientos a realizar para la implementación del sistema, por otro lado, la documentación generada por metodologías convencionales de desarrollo de software reflejarían información con poco o ningún valor de utilidad debido a que el SBC se centrará esencialmente en elementos algorítmicos, matemáticos y de programación.

1.6.2. Lenguaje de programación

Para el desarrollo del sistema de recomendación se analizaron los lenguajes PHP y Java por ser en la actualidad, dos delos lenguajes más utilizados a nivel mundial(41), (42) y porque son los dos lenguajes que han estudiado los desarrolladores. A continuación se especifican las características más importantes de estos lenguajes de programación.

PHP

PHP, acrónimo recursivo que significa PHP *Hypertext Pre-processor*, fue creado originalmente por Rasmus Lerdorf en 1994. Es un lenguaje de alto nivel específicamente pensado para desarrollar páginas web por lo cual puede ser incrustado en páginas HTML. Fue liberado bajo la PHP *License; la Free Software Foundation* (FSF) considera como software libre a los productos liberados bajo esta licencia. Entre sus características se puede señalar que es un lenguaje que consume pocos recursos, por lo que se ejecuta con rapidez y no tiende a frenar el resto de los procesos del sistema operativo. Además tiene gran capacidad de conectividad, al utilizar un sistema de extensiones modular hacia interfaces de variado tipo, tales como librerías gráficas, XML, encriptación. Entre sus desventajas se pueden mencionar que para la realización de proyectos complejos requiere un alto nivel de experiencia. También como es un lenguaje que se interpreta en ejecución para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. La ofuscación es una técnica que puede dificultar la lectura del código pero no la impide y, en ciertos casos, representa un costo en tiempos de ejecución.(43), (44)

Java

Java surgió en 1991 cuando un grupo de ingenieros de Sun Microsystems trató de diseñar un nuevo lenguaje de programación destinado a correr sobre cualquier plataforma: desde computadoras hasta efectos electrodomésticos. Con este fin fueron de los primeros en introducir el concepto de máquina virtual con el objetivo de que sus aplicaciones fueran totalmente independientes del CPU que las procesara y esto constituye una ventaja, ya que, corriendo sobre la máquina virtual, te permite olvidarte de algo tan engorroso como es la gestión de memoria (punteros, reserva, etc.). (45), (46)

Sun describe a Java como simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico. Se ha desarrollado unido a Internet y es por ello que es ampliamente utilizado en la construcción tanto de sistemas distribuidos, como de complejas aplicaciones de gestión. Otra de sus ventajas es que cuenta con una librería de clases bastante completa que se puede aprovechar para el desarrollo de aplicaciones para distintas tecnologías (de escritorio, móvil, web).(45), (46),(47)

Entre las limitaciones de Java se puede mencionar su velocidad, los programas hechos en Java no tienden a ser muy rápidos. Como los programas de Java son interpretados nunca alcanzan la velocidad de un verdadero ejecutable. Otra desventaja que posee Java es que algunas implementaciones y librerías pueden tener código rebuscado. (48)

Fundamentación de la elección

Se decide el uso de Java principalmente por las siguientes razones:

- Java tiene librerías que serán útiles a los desarrolladores para realizar la recomendación y para la comparación de String, permitiendo así, minimizar tiempo y esfuerzo en el desarrollo del sistema.
- Los desarrolladores tienen un mayor conocimiento de java y poseen una mayor experiencia en el desarrollo de software en este lenguaje.

1.6.3. Gestor de base de datos

Un **sistema de gestión de bases de datos** (**SGBD**) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos.(49)Para la selección

del gestor de la base de datos (BD) de la aplicación fueron analizados MySQL y PostgreSQL, por contener como parte de su desarrollo versiones que constituyen tecnología libre.

MySQL

Es un sistema de gestión de bases de datos relacional.

Características de MySQL(50), (51):

- Aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

Ventajas (50), (51):

- MySQL software es de código abierto.
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que por su bajo consumo puede ser ejecutado en una máquina con escasos recursos.
- Facilidad de configuración e instalación. Soporta gran variedad de Sistemas Operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.

Desventajas(50), (51):

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es viable para su uso con grandes bases de datos, a las que se acceda continuamente.

PostgreSQL

Es un sistema de gestión de bases de datos basado en el proyecto POSTGRES, de la universidad de Berkeley. PostgreSQL utiliza el lenguaje SQL92/SQL99. Es un sistema objeto-relacional, ya que incluye

características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.(50), (51):

Características de PostgreSQL(50), (51):

- Soporta distintos tipos de datos: datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios.
- Incorpora una estructura de datos *array.*
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

Ventajas(50), (51):

- Licencia gratuita.
- Ahorros considerables en costos de operación.
- Postgresql ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características, estabilidad y rendimiento.
- Herramientas gráficas de diseño y administración de BD.
- No se han presentado caídas de la base de datos.
- Puede operar sobre distintas plataformas, incluyendo Linux, Windows, Unix, Solaris y MacOS X.
- Gran capacidad de almacenamiento.

- Buena escalabilidad ya que es capaz de ajustarse al número de CPU y a la cantidad de memoria disponible de forma óptima, soportando una mayor cantidad de peticiones simultáneas a la base de datos de forma correcta.

Desventajas (50), (51):

- Consume más recursos que MYSQL por lo que se necesitan mayores características de hardware para ejecutarlo.
- La sintaxis de algunos de sus comandos o sentencias no es nada intuitiva.

Fundamentación de la elección

A pesar de que MySQL es más rápido que PostgreSQL a la hora de resolver consultas, no es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad. PostgreSQL ofrece una garantía de integridad mucho más fuerte que MySQL. Además, presenta una mejor escalabilidad y rendimiento bajo grandes cargas de trabajo. También corre en los sistemas operativos más extendidos como Linux, Unix, Windows, BSDs, Mac OS y otros. Tiene una documentación muy bien organizada y se adapta fácilmente a las necesidades del cliente. Soporta todas las características de una base de dato profesional (funciones, secuencias, *tiggers*, relaciones, reglas, tipo de datos definido por el usuario, vistas, etc.).

Por tanto, se decide usar PostgreSQL para el desarrollo de la aplicación.

1.7. Conclusiones parciales

El análisis de las características de los REA y de los patrones de diseño que se utilizan en la creación de estos recursos, permitió comprender mejor el problema de investigación. Además, el estudio de los sistemas de recomendación desde su evolución, sus principales elementos y clasificación posibilitó la selección del razonamiento basado en casos para el desarrollo de la aplicación seleccionando dentro de su ciclo de vida las fases de recuperación y reutilización de casos. El lenguaje Java y el gestor de base de datos PostgreSQL elegidos permitieron el desarrollo de la solución propuesta.

Capítulo 2. Concepción del sistema recomendador basado en casos

Introducción

En este capítulo se explica en qué consiste el sistema recomendador de patrones para REA (REPREA) que se va a desarrollar. Se describen las etapas de recuperación y reutilización de casos del ciclo CBR, explicando las técnicas que se utilizan en la comparación de los casos. Además, se define los patrones arquitectónicos y de diseño que serán utilizados para el desarrollo de la aplicación.

2.1. CBR Textual

El CBR Textual es un subapartado del CBR donde las fuentes del conocimiento se dan en formato de texto. El objetivo es utilizar este conocimiento textual para resolver un problema de la misma manera que el CBR, es decir, comparando la consulta inicial en formato textual, con la base de casos (también en formato textual total o parcialmente) y recuperar los más similares. Se mantiene, por tanto, el mismo esquema del ciclo CBR: recuperar, reutilizar, revisar y recordar.(52), (53), (54)

Esta rama del razonamiento basado en casos plantea varias cuestiones: cómo hacer similitud entre casos textuales, cómo adaptar casos textuales o cómo generar automáticamente representaciones textuales de los casos. Estas y otras cuestiones se resuelven con técnicas de recuperación de información. Estas técnicas trabajan con textos no estructurados, consultas en lenguaje natural y realizan una recuperación aproximada.(52), (53), (54)

A continuación se describen las etapas del ciclo de CBR con los métodos y algoritmos utilizados en cada una de ellas para darle solución al problema planteado en esta investigación.

2.1.1. Fase Recuperación de casos

Para la recuperación de casos se utiliza la recuperación de información (RI), definida como el problema de la selección de información depositada en un medio de almacenamiento, en respuesta a consultas realizadas por un usuario. Teniendo en cuenta el aumento de la disponibilidad de documentos en soporte electrónico y la necesidad de obtener en cada momento aquellos que responden a una necesidad informativa, es que surge este concepto. (55), (56)

Cualquier sistema que utilice la RI puede ser descrito como un conjunto de ítems de información, un conjunto de peticiones y algún mecanismo que determine que ítem satisface las necesidades de

información expresadas por el usuario en la petición, guiándose por la similitud existente entre la consulta del usuario y los ítems de información.(55), (56)

Los sistemas que utilizan la RI son sistemas que tratan con bases de datos compuestas por documentos y procesan las consultas de los usuarios permitiéndoles optimizar el tiempo de acceso a la información que estos necesitan. Estas consultas son sentencias formales mediante las cuales el usuario expresa sus necesidades.(55), (56)

Funciones principales de la RI:(56),(57)

- Identificar las fuentes de información relevantes a las áreas de interés de las solicitudes de los usuarios.
- Analizar los contenidos de los documentos.
- Representar los contenidos de las fuentes analizadas de una manera adecuada para compararlas con las preguntas de los usuarios.
- Analizar las preguntas de los usuarios y representarlas de una forma que sea adecuada para compararlas con las representaciones de los documentos de la base de datos.
- Realizar la correspondencia entre la representación de la búsqueda y los documentos almacenados en la base de datos.
- Recuperar la información relevante.
- Realizar los ajustes necesarios en el sistema basados en la retroalimentación con los usuarios.

Para entender mejor estas funciones, se presenta la siguiente figura:

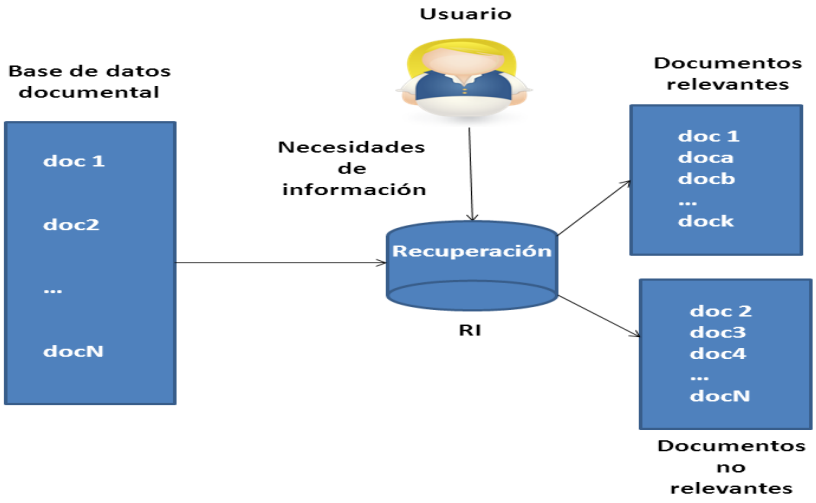


Figura 7. Proceso de recuperación de información

Después de haber estudiado los elementos principales de la recuperación de información, se procede a explicar cómo se realiza en el caso de REPREA, tomando en la base de casos a los patrones como los documentos de la base documental.

La tarea de recuperación de casos comienza con una descripción del problema, y finaliza cuando se encuentran un conjunto de casos semejantes al caso nuevo que se desea resolver. Se divide en dos pasos: identificación de características y emparejamiento.

La tarea de identificación de características del problema consiste en proporcionar un conjunto de descriptores relevantes del problema y el objetivo de la tarea de emparejamiento es devolver un conjunto de casos suficientemente similares al nuevo caso dado un criterio de similitud de algún tipo.(36),(58)

Identificación de características

La identificación de características del problema puede simplemente consistir en el reconocimiento de sus descriptores de entrada. Se intenta comprender el problema dentro de su contexto. (36),(58), (59)Para esto se utilizará la recuperación de información (RI) basada en la técnica de n-gramas contextuales.

N-gramas Contextuales: Reciben este nombre por su capacidad de representar la esencia del contexto con un reducido número de caracteres. Son el resultado de agrupación de *n* palabras en un texto, siendo *n* la cantidad de palabras que se decidan agrupar, después de la previa eliminación de las palabras vacías y caracteres aislados, extracción del lexema (*stem*) y ordenación interna de cada n-grama. Estudios anteriores sobre la conveniencia del grado de n-gramas manifestaron que los bigramas y trigramas son las mejores opciones. Se demostró, además, que el mejor resultado se obtenía mediante n=3, con significativa mejor cobertura, precisión y granularidad que mediante el empleo de n=2.(60), (61), (62)

Modelar con n=3, permite desambiguar los bigramas mediante la palabra inmediatamente anterior y posterior. Los trigramas no toman estas dos únicas referencias posibles, sino que las amplían a las palabras pre-anterior y post-siguiente al bigrama, consiguiendo que su identificación no dependa exclusivamente de la monotonía de conservación del texto original, triplicando las oportunidades para su identificación.(60), (61), (62)

Para conseguir que el n-grama contenga la mejor definición de la esencia del contexto y sea especialmente útil para comparar dos textos, se llevan a cabo cinco pasos(60), (61), (62):

1. La conversión a minúsculas es una práctica común.

2. La eliminación de las palabras vacías conocidas también como *stopwords*.

Las palabras vacías son palabras que carecen de significado. Dentro de esta categoría se pueden incluir artículos, conjunciones y pronombres. La eliminación de estas palabras, por lo general, mejora la relevancia de los resultados obtenidos, ya que al no tenerlas en cuenta, no se encuentran falsas coincidencias, además, esta eliminación de palabras vacías incrementa el rendimiento al aligerar las consultas y eliminar datos irrelevantes. Esta técnica hace que el n-grama contenga una información mucho más definitoria del contexto de la sentencia analizada.(62),(63)

3. Eliminación de palabras de un solo caracter. Este paso permite que no se afecte la comparación entre los textos ante el posible cambio de orden en la cadena por ejemplo cuando se hace uso de conjunciones como “y” y “o”.

4. La reducción a la raíz *(stemming)* 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.

Una vez analizados estos pasos, se presenta un ejemplo de extracción de n-gramas contextuales de la siguiente frase: “En un lugar de la mancha de cuyo nombre no quiero acordarme”, se extraen los siguientes n-gramas con n=3:

lugar_mancha_nombre mancha_nombre_querer acordar_nombre_querer lugar_mancha_querer acordar_mancha_nombre lugar_nombre_querer acordar_mancha_querer

A continuación, se procede a explicar cómo se garantizó la ejecución de cada uno de estos cinco pasos para comparar los casos textuales en el desarrollo de REPREA. Para algunos casos se reutilizaron funciones de Java que se encuentran implementadas y disponibles para los desarrolladores y que se ajustan a los resultados esperados; en otros casos se realizaron las implementaciones necesarias.

Para el primer paso, la conversión a minúsculas, se utilizó la función de comparación de String en java equalsIgnoreCase()(64),(65), que compara sin tener en cuenta las mayúsculas. Después, se hizo una eliminación de las palabras vacías y de las de un solo caracter con el método eliminarPalabra(String s)

donde se recorre la cadena de texto y se eliminan los pronombres, artículos, preposiciones y las palabras de un solo caracter, de forma tal, que cuando se vaya a comparar, estas no influyan en el valor de semejanza entre los textos.

Algoritmo 1.Eliminar palabras vacías y de un solo caracter.

***Entrada**:* una cadena de texto (el problema de REA).

***Salida**:* una lista de *string* (guarda las palabras del problema de REA sin incluir las palabras vacías y sin las de un solo caracter).

Variable: s {es el problema de REA entrado al sistema}, arreglo {las palabras del problema entrado, cada una en una posición}, a {es una lista con las palabras del problema de REA sin incluir las palabras vacías y sin las de un solo caracter}, prepos {es un arreglo de preposiciones y pronombres que tienen una longitud mayor que 4}

Inicio

Arreglo ←s.split(" ") {Se guardan en las posiciones del arreglo las palabras del problema de REA mediante el método Split que separa las palabras del texto cada vez que encuentra un espacio en blanco}

Prepos ← {es un arreglo de preposiciones y pronombres que tienen una longitud mayor que 4}

a← lista de *string* que se va a devolver sin incluir las palabras vacías y sin las de un solo caracter

Long ← arreglo.longitud() {leer la longitud del arreglo}

{Comienza el ciclo}

Para i←0,

mientras *que i<long* ***hacer***

> ***Si*** *se cumple Arreglo[i].longitud() > 4*{se toma la posición i del arreglo que es una palabra y si la cantidad de caracteres que conforman esa palabra es mayor que 4} ***entonces***
>
> *a.añadir(arreglo(i))* {se añade la palabra a la lista a, con esto se eliminan todas las palabras que tengan una longitud menor que 4}
>
> ***fin del si***

i←i+1 {se incrementa i}

fin del ciclo

{Comienza el ciclo 1}

Para i ←0,

mientras *i<a. longitud()* {i menor que la longitud de la lista}***hacer***

> ***Comienza el ciclo 2*** {es un doble ciclo para que por cada palabra de la lista se haga lo siguiente}
>
> *Para i←0*;
>
> ***Mientras*** *i< prepos.longitud()* {si i es menor que la longitud del arreglo de preposiciones y pronombres} ***hacer***

Si *a.Obtener(i)== prepos[j]* {se compara la palabra que está en la posición i de la lista con todas las preposiciones y pronombres que hay en el arreglo prepos, en caso que alguna palabra sea una preposición o pronombre} **entonces**

a.remover(i){se elimina esa palabra de la lista}

Fin del si

i←i+1 {se incrementa i}

Fin del ciclo 2

i←i+1 {se incrementa i}

Fin del ciclo 1

retornar*a* {se devuelve la lista *a* con las palabras de la cadena de texto sin las palabras vacías y sin las de un solo caracter}

Fin del método

Una vez eliminadas las palabras vacías y las de un solo caracter, hay que extraer el lexema de las palabras. El lexema es una parte que constituye la unidad mínima de una palabra y se puede decir que es la raíz de esta misma, además es la parte que se mantiene invariable en todas las palabras de una misma familia; expresa el significado común a toda la familia y puede coincidir o no con una palabra entera ejemplo: el lexema de pato es “pat”; el lexema de lápiz es “lápiz”.

Para la extracción del lexema se utiliza la clase Stemm_es (66)donde se reduce la palabra a su raíz como por ejemplo: educativo se reduce a “educa”, para cuando se vaya a comparar las frases “Crear un recurso educativo abierto para estudiantes de primaria” y “Crear un recurso para educar a los estudiantes de primaria” las palabras educativo y educar cuando se extraiga su lexema que sería “educa” y se comparen son iguales completamente.

Por último, para la combinación de los n-gramas del texto a analizar se implementó el método CombinarNgramas(String c).

Algoritmo 2. Combinar n-gramas de una cadena de texto

Entrada*: una cadena de texto* (el problema de REA).

Salida*: Lista de string* (conjunto de n-gramas de la cadena de texto).

Variables*:* nue {es una lista de *string* que guarda la lista de string que devolvió el método de eliminarPalabras}, listadengramas {lista con el conjunto de n-gramas que se va a devolver}

Inicio

listadengramas ← *new LinkedList<String>()* {se inicializa la lista de *string* que se va a devolver con el conjunto de n-gramas}

nue ← *eliminarPalabra(cadena)* {se hace un llamado al método de eliminar palabras vacías y de un solo caracter pasándole el problema de REA y se guarda la lista en *nue* }

{Comienza el ciclo}

Para i←0

mientras i< *nue.longitud()-2* {mientras que i sea menor que el tamaño de la lista menos 2, tiene que ser menos dos porque cuando se forman los n-gramas hay que coger la posición i junto con i+1 y con i+2 entonces si el ciclo llega hasta el final cuando diga i+2 esa posición no la va a encontrar} ***hacer***

ngrama←nue.Obtener(i) + "_" + nue.Obtener(i + 1) + "_" + nue.Obtener(i + 2) {se recorre la lista *nue* de palabras y se unen tres posiciones por un guión bajo y así se forman los n-gramas}

listadengramas.añadir(ngrama) {se añade el n-grama a la lista que se va a devolver}

i←i+1 {se incrementa i}

Fin del ciclo

Comienza el ciclo {este es para formar más n-gramas pero de otra combinación}

Para i←0,

Mientras *i< nue.longitud()-3* {mientras que i sea menor que el tamaño de la lista menos 3, tiene que ser menos tres porque cuando se forman los n-gramas hay que coger la posición i junto con i+1 y con i+3 entonces si el ciclo llega hasta el final cuando diga i+3 esa posición no la va a encontrar} ***hacer***

ngrama←nue.Obtener(i) + "_" + nue.Obtener(i + 1) + "_" + nue.Obtener(i + 3) {se recorre la lista *nue* de palabras y se unen tres posiciones por un guión bajo y así se forman los n-gramas}

listadengramas.añadir(ngrama) {se añade el n-grama a la lista que se va a devolver}

i←i+1 {se incrementa i}

Fin del ciclo

Comienza el ciclo {este es para formar más n-gramas pero de otra combinación, en este caso se recorre la lista desde el final hacia el inicio}

Para i←nue.longitud()-1, *i>=3*{para i igual a la longitud de la lista menos 1 por lo mismo que se ha explicado anteriormente de la posición no encontrada y siendo i siempre mayor o igual que 3} ***hacer***

ngrama←nue.obtener(i) + "_" + nue.obtener(i - 2) + "_" + nue.obtener(i - 1){se recorre la lista *nue* de palabras y se unen tres posiciones por un guión bajo y así se forman los n-gramas pero desde la última posición hasta la primera de la lista}

listadengramas.añadir(ngrama) {se añade el n-grama a la lista que se va a devolver}

ngrama1←nue.obtener(i) + "_" + nue.obtener(i - 3) + "_" + nue.obtener(i - 1) {otra combinación de n-grama}

listadengramas.añadir(ngrama1) {se añade el n-grama a la lista que se va a devolver}

i←*i-1*{se disminuye i}

Fin del ciclo

retornar*listaden-gramas* {se devuelve la lista de n-gramas con todos los n-gramas formados de la cadenas de texto}

Fin del método

Emparejamiento

La tarea de emparejamiento es un proceso en el cual se recupera un conjunto de posibles candidatos.(36), (58), (59)La búsqueda de un conjunto de casos posibles semejantes al problema que se desea resolverse hace usando los descriptores del problema (características de entrada) como índices a la memoria de casos.(36), (58), (59)

La recuperación o selección de los casos de la base de conocimiento semejantes al problema actual se puede realizar utilizando dos métodos(58), (59), (67):

- recuperación por semejanza parcial
- recuperación por analogía

En la recuperación por semejanza parcial se emplea una función de semejanza. Esta función, al ser calculada, se obtiene una medida numérica, la cual representa el grado de similitud de cada texto de la base de casos con respecto al nuevo problema, por ejemplo(68), (69):

$$\boldsymbol{P, C} \;=\; \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{w_i} * \boldsymbol{\delta_i}\ \boldsymbol{P_i, C_i}$$

- Donde $\boldsymbol{w_i}$ es la importancia del rasgo,$\boldsymbol{P_i}$y$\boldsymbol{C_i}$son los valores que el rasgo $\boldsymbol{i}$tiene en el problema y en el caso respectivamente, y $\boldsymbol{\delta_i}$es la función de comparación para el rasgo $\boldsymbol{i.}$(68), (69)

$$\boldsymbol{f_e^i} = \frac{\boldsymbol{x_p\ \ 0_i\ \ - x_p^0}}{\boldsymbol{x_p^{l(p)} - x_p^0}}$$

- Donde$\boldsymbol{f_e^i}$esun coeficienteen el cual el numerador es la diferencia entre el valor que toma el rasgo$\boldsymbol{p}$en el objeto $\boldsymbol{i}$y el menor valor posible para ese rasgo, y el denominador es la máxima variación posible de los valores del rasgo $\boldsymbol{p}$. Se considera que los elementos del rango $\boldsymbol{i}$tienen un orden, $\boldsymbol{x_p^0}$denota el primer elemento y $\boldsymbol{x_p^{l(p)}}$ el último.(68), (69)

Un ejemplo de problema que se resuelve con este tipo de recuperación es el siguiente(68), (69):

Se desea determinar si un paciente presenta o no una cardiopatía. Para esto es necesario tener en cuenta los siguientes aspectos: si el paciente fuma (sí, no); la toma de bebidas alcohólicas (no, poco, mucho), el uso de fármacos contraindicados (sí, no, no sé), presión arterial mínima (50-100unidades), presión arterial máxima (80, 200 unidades). Determine si un paciente con la siguiente descripción (Paciente1)= (sí, mucho, no sé, 80, 120) presenta o no una cardiopatía. Cada uno de los aspectos descritos serían los rasgos a tener en cuenta para calcular la función de semejanza.

A continuación, se muestra en la tabla la base de casos de diferentes pacientes ya analizados anteriormente para darle solución al problema planteado:

Tabla 1. Rasgos

ID	Fuma	Bebidas alcohólicas	Fármacos contraindicados	PA mínima	PA máxima	Presenta cardiopatía
1	Sí	No	No	80	120	No
2	No	Poco	no sé	70	110	No
3	No	Mucho	Sí	100	160	Sí
4	Sí	Poco	Sí	90	130	No
5	No	Mucho	No	80	120	Sí

En la recuperación por analogía, se buscan los textos de la base de casos cuya descripción pueda hacerse igual o semejante a la del nuevo problema.(58), (59), (67)

Fundamentación de la elección

Para la recuperación de casos en REPREA se utiliza el método de recuperación por analogía, pues en el caso del método por semejanza parcial, para calcular las funciones hace falta tener identificados una serie de rasgos del problema que se quiere resolver y la base de casos está estructurada a partir de rasgos ya predefinidos. En el caso de la presente investigación, se dispone de una base de casos de patrones de diseño de REA descritos en forma de texto y la descripción incluye el problema que resuelve y la solución a este, además, la entrada que tendrá REPREA es un texto que describe un problema de un REA. Por tanto, el método que mejor se ajusta a lo planteado es el de recuperación por analogía para comparar la descripción del problema de diseño del REA que se desea resolver, con la descripción de los problemas

que hay almacenados en la base de casos de patrones se seleccionará, de esta forma, el o los más semejantes.

Para realizar esta comparación textual, se utilizará la técnica de los n-gramas contextuales, ya analizada anteriormente. Con el objetivo de determinar si el problema entrado *s* puede ser semejante al problema almacenado en la base de casos *d* se han propuesto dos medidas principales: semejanza (S) y contención(C).(61), (70)

La medida de semejanza es útil cuando los conjuntos de n-gramas a comparar provienen de textos de longitud equiparable. Considerando un problema *s* y uno de la base de casos *d*, la semejanza se define por medio de la ecuación:

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

Donde *N(d)* es el conjunto de n-gramas en la cadena de texto *d* de la base de casos y *N(s)* es el conjunto de n-gramas en la cadena de texto *s* que será entrada al sistema.(61), (70), (71)

A continuación se explica el método Semejanza entre dos cadenas de textos donde se utiliza la ecuación antes descrita.

Algoritmo 3. Semejanza

***Entrada**:* dos cadenas de textos

Salida: un número entero que si es mayor o igual a 0.8 son semejantes las cadenas de textos

Variables: semejanza (resultado al calcular la fórmula), l (lista de n-gramas de la cadena 1), li (lista de n-gramas de la segunda cadena), coj (conjunto de ngramas de cadena 1), coji (conjunto de ngramas de cadena 2), inter (resultado de la intersección entre las dos listas), unión (resultado de la unión entre los dos conjuntos de ngramas)

Inicio

Semejanza ←0

l ←extraerNgramas(cadena 1){se guardan los n-gramas de la cadena 1 en la lista l}.

li ←extraerNgramas(cadena 2){se guardan los n-gramas de la cadena 2 en la lista li}.

coj ←Conjunto(l) {se crea un conjunto con la lista l}

coji ←Conjunto(li) {se crea un conjunto con la lista li}

inter ← coj.interseccion(l,li) {el método intersección devuelve el número de n-gramas que son iguales en las dos listas}.

unión←coj.union(coji) {el método unión devuelve un conjunto nuevo *S* con todos los n-gramas de los dos conjuntos coj y coji sin repetición de n-gramas}.

Si *(inter/unión.getS.length) >=0.8*{si se cumple que la intersección de las dos listas de n-gramas dividido entre la cantidad de elementos que tiene el conjunto *S* (que es la unión de los n-gramas de las dos cadenas de textos) es mayor o igual a 0.8}***entonces***

Semejanza→1

Fin del si

retornar *semejanza*

fin del método

En caso de que los textos que se deseen comparar no tengan una longitud equiparable, la opción es la medida de contención, la cual se define en la ecuación:

$$C(si \mid d) = \frac{|N(si) \cap N(d)|}{|N(si)|}$$

donde *si* es el conjunto de n-gramas en la cadena de texto *si* que será entrada al sistema.(61), (70), (71)

A continuación se explica el método Contención entre dos cadenas de textos donde se utiliza la ecuación antes descrita.

Algoritmo 4. Contención

Entrada*:* dos cadenas de textos

Salida*:* un número entero que si es mayor o igual a 0.5 está contenida una cadena dentro de la otra.

Variables: contención (resultado al calcular la fórmula), l (lista de n-gramas de la cadena 1), li (lista de n-gramas de la segunda cadena), conj (conjunto de n-gramas de cadena 1), inter (resultado de la intersección entre las dos listas)

Inicio

contencion←0

l←extraerNgramas(cadena 1) { se guardan los n-gramas de la cadena 1 en la lista l}

li←extraerNgramas(cadena 2) {se guardan los n-gramas de la cadena 2 en la lista li}

conj←Conjunto(l) {se crea un conjunto con la lista l}

inter← coj.interseccion(l,li) {el método intersección devuelve el número de n-gramas que son iguales en las dos listas}.

Si *(inter/conj.getS.length) >=0.5* {si se cumple que la intersección de las dos listas de n-gramas dividido entre la cantidad de elementos que tiene el conjunto *S* (que es el conjunto de n-gramas de la cadena 1) es mayor o igual a 0.5} ***entonces***

contencion→1

fin del si

retornar contencion

fin del método

Tanto la semejanza como la contención son valores dentro del intervalo [0, 1]. (71), (72) Es necesario definir un umbral dentro de este intervalo tal que al ser superado, se considere que el texto entrado por parámetro es semejante al texto de la base de casos, o esté contenido en él dependiendo de la ecuación que se utilice.(71)El umbral definido para la investigación es de 0.8 para la semejanza y para la contención es de 0.5.(71), (72)

2.1.2. Fase Reutilización de casos

Después de haber realizado una búsqueda más o menos exhaustiva sobre el conjunto de casos para extraer aquellos casos que presentasen más características en común, ha llegado el punto en que será necesario seleccionar de este conjunto de candidatos aquel o aquellos casos más propensos a ser una buena solución para el problema.

El proceso de elección es más elaborado que el anterior, principalmente porque en el anterior, el conjunto de casos que se tenía que comprobar era mayor al actual y con un conjunto de casos más pequeño es posible realizar pruebas más exhaustivas. En el proceso de selección los casos son puntuados según su nivel de similitud con el problema inicial, permitiendo ordenarlos para resultar futuros candidatos finales. Otros criterios que se utilizan para la selección es el discriminar el hecho de que alguna característica determinada no sea igual.(58)

Es importante que los casos resultantes de la selección sean lo suficientemente similares a los objetivos requeridos por el problema, ya que si el proceso no es todo lo adecuado que se desea, en siguientes pasos los resultados que se obtendrían no serían tan buenos como los esperados.

La reutilización de los casos tiene mucha importancia ya que con una correcta implementación de esta técnica se mejorará la forma en que se aprovechan las características que tienen los casos y por lo tanto, se recomendará una mejor solución final al problema.(58)

La reutilización de los casos recuperados se fundamenta básicamente en dos aspectos(58):

- Las diferencias entre el caso recuperado de la base de conocimientos y el nuevo caso.
- La parte del caso recuperado de la base de conocimiento que puede ser utilizado en el nuevo caso.

El algoritmo implementado para seleccionar el caso o los casos que más se ajustan al problema entrado al sistema se llama SimilitudEntreCadenas(String a,String b), este método es aplicado a los casos recuperados en la fase anterior con las ecuaciones de semejanza y contención.

Algoritmo 5. Similitud entre dos cadenas

Entrada: las dos cadenas que se quieren comparar

Salida: *un numero entero 0, 1, 2* {0 significa que son iguales completamente, 1 que la primera cadena es subcadena de la segunda, 2 que la segunda cadena es subcadena de la primera}

Las cadenas que se van a comparar en esta investigación serían: el problema de REA entrado al sistema y el problema de REA que está almacenado en el patrón de diseño en la base de casos de patrones. Por tanto, cuando el método devuelva 0, el patrón que resuelva el problema que se está comparando en ese momento será el recomendado al diseñador para que pueda resolver el problema de diseño de REA que entró al sistema.

Variable*:* objeto {es un objeto de la clase Stemm}, Lista {lista con las palabras de la primera cadena}, lista1 {lista con las palabras de la cadena 2}, Lis {lista con las raíces de las palabras de la cadena 1 } , Lis1 {lista con las raíces de las palabras de la cadena 2}, contador { un número entero}

Inicio

Contador←0

{Comienza el ciclo}

Para i←0

mientras *i<lista.longitud()* {mientras hayan palabras en la lista} ***hacer***

String m←objeto.Stem(lista.obtener(i)) {se pasan todos los elementos de la lista que son las palabras de la primera cadena por el método stem que lo que hace es extraer la raíz de las palabras}

Lis.añadir(m){se añaden a la lista las raíces de las palabras de la cadena 1}

i←i+1 {se incrementa i}

Fin del ciclo

{Comienza el ciclo}

Para i←0

mientras *i<lista1.longitud()* {mientras hayan palabras en la lista}***hacer***

String m←objeto.Stem(lista1.obtener(i)) {se pasan todos los elementos de la lista 1 que son las palabras de la segunda cadena por el método stem que lo que hace es extraer la raíz de las palabras}

Lis1.añadir(m) {se añaden a la lista las raíces de las palabras de la cadena 2}

i←i+1 {se incrementa i}

Fin del ciclo

{Inicio del ciclo}

Para i←0

Mientras *i< lis.longitud()****hacer***

{comienza el ciclo}

Para j←0

Mientras *j< lis1.longitud()hacer*

Si *(lis.obtener(i).equalsIgnoreCase(lis1.obtener(j)))* { se van comparando las palabras de las dos cadenas manteniendo las palabras de la cadena 1 fija por ejemplo: se compara la primera palabra de la primera lista (que son las de la cadena 1) con todas las de la segunda cadena y cuando termina se coge la posición dos de la primera cadena se mantiene fija otra vez y se compara con todas las palabras de la segunda cadena y así sucesivamente hasta terminar con todas las palabras, y cada vez que sean iguales} ***entonces***

Contador←contador+1 {se incrementa el contador};

j←j+1 {se incrementa j};

fin del ciclo

i←i+1 { se incrementa i}

fin del ciclo

Si(contador es igual la longitud de la lista de las palabras de la cadena 1 y a la longitud de la lista de las palabras de la cadena 2) o (contador es igual la longitud de la lista de las palabras de la cadena 1 y a la longitud de la lista de las palabras de la cadena 2 menos 1) o (contador es igual a la longitud de la lista de las palabras de la cadena 1 menos 1 y a la longitud de la lista de las palabras de la cadena 2) cualquiera de esas combinaciones significa que la mayoría de las palabras son iguales excepto una en los dos últimos casos, si se cumplen cualquiera de las tres condiciones ***entonces***

retorna 0{significa que las cadenas son iguales}

fin del si

Si (contador es igual la longitud de la lista de las palabras de la cadena 1 y es distinto de la longitud de la lista de las palabras de la cadena 2) ***entonces***

retorna 1 {la primera cadena pertenece a la segunda}

fin del si

Si (contador es distinto de la longitud de la lista de las palabras de la cadena 1 y es igual a la longitud de la lista de las palabras de la cadena 2) ***entonces***

retorna 1 {la segunda cadena pertenece a la primera}

fin del si

sino {se cumple ninguna de las condiciones anteriores}

retorna-1 {no hay similitud entre las cadenas}

fin del método

2.2. Patrones utilizados

2.2.1. Patrón arquitectónico

La rama de la ingeniería de software se preocupa por crear procesos que aseguren calidad en los programas que se realizan y esa calidad atiende a diversos parámetros que son deseables para todo desarrollo, como la estructuración de los programas o reutilización del código, lo que debe influir positivamente en la facilidad de desarrollo y el mantenimiento. (73)

Para asegurar la calidad en el software aparecen los llamados patrones arquitectónicos. Estos se utilizan para expresar una estructura de organización base o esquema para un software. Proporcionando un conjunto de sub-sistemas predefinidos, especificando sus responsabilidades, reglas, directrices que determinan la organización, comunicación, interacción y relaciones entre ellos.(74),(75)

Los patrones arquitectónicos heredan mucha de la terminología y conceptos de patrones de diseño, pero se centran en proporcionar modelos y métodos re-utilizables específicamente para la arquitectura general de los sistemas de información. Lo que significa que a diferencia de los patrones de diseño, estos son plantillas incompletas y no se pueden aplicar directamente al código con modificaciones meramente contextuales. Los patrones arquitectónicos, a su vez, se salen del código puro de la aplicación y suben e incluyen software, hardware, redes, inclusos las personas.(73),(74)

Los ingenieros de software se dedican a estudiar de qué manera se pueden mejorar los procesos de creación de software y una de las soluciones a las que han llegado es la arquitectura basada en capas que separan el código en función de sus responsabilidades o conceptos. Un ejemplo de este tipo de arquitectura es el patrón Modelo-Vista-Controlador (MVC), que permite a los desarrolladores de software crear aplicaciones con mayor calidad. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.(73)

MVC separa los datos de una aplicación en tres componentes distintos:(73),(76),(77)

- La capa del **modelo** define la lógica de negocio. Es la capa donde se trabaja con los datos, por tanto contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos se tendrán habitualmente en una base de datos, por lo que en los modelos se tendrá todas las funciones que accederán a las tablas y harán los correspondientes *selects, updates, inserts*, etc.
- La **vista** es lo que utilizan los usuarios para interactuar con la aplicación.
- El **controlador** es un bloque de código que realiza llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario. Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, una búsqueda de información. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo.

Flujo de control: (76)**,**(77)

1. El usuario realiza una acción en la interfaz

2. El controlador trata el evento de entrada.

3. El controlador notifica al modelo la acción del usuario, lo que puede implicar un cambio del estado del modelo (si no es una mera consulta).

4. Se puede generar una nueva vista. La vista toma los datos del modelo.

- El modelo no tiene conocimiento directo de la vista

5. La interfaz de usuario espera otra interacción del usuario, que comenzará otro nuevo ciclo.

Dos ejemplos que demuestran las ventajas de utilizar este patrón son los siguientes:(73)

1. Si se desea que en un equipo intervengan perfiles distintos de profesionales y trabajen de manera autónoma, como diseñadores o programadores, y no se ha utilizado el patrón MVC, ambos tienen que tocar los mismos archivos y el diseñador se tiene que relacionar con mucho código en un lenguaje de programación que puede no serle familiar y a él solo le interesa trabajar con las vistas al usuario, igual pasa con el programador, necesita trabajar en la implementación de las clases controladoras, sin tener necesidad de trabajar con el diseño de la aplicación. Por eso, sería mucho más fácil la separación del código utilizando MVC.

2. Durante la manipulación de datos en una aplicación es posible que se esté accediendo a los mismos datos en lugares distintos. Por ejemplo, se puede acceder a los datos de un artículo desde la página donde se muestra éste, la página donde se listan los artículos de un manual o la página de donde se administran los artículos de un sitio web. Si un día se cambian los datos de los artículos (se modifica la tabla para añadir nuevos campos o cambiar los existentes porque las necesidades de los artículos varían), se obliga a cambiar, página a página, todos los lugares donde se consumían datos de los artículos. Además, si se tiene el código de acceso a datos disperso por decenas de lugares, es posible que se repitan las mismas sentencias de acceso a esos datos y por tanto no se reutiliza código.

Por tanto, para el desarrollo de REPREA se utiliza el patrón arquitectónico MVC teniendo como modelo la base de casos de patrones de diseño, como controlador, una clase controladora que tiene implementada las funcionalidades necesarias para satisfacer las necesidades de desarrollo y responder a las acciones que se soliciten en la aplicación y como vista se tiene un conjunto de formularios que permiten la interacción de los diseñadores de REA con el sistema.

2.2.2. Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. (78), (79), (80)

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares.(78), (79), (80)

Patrones GRASP

Son patrones que resaltan la importancia de captar principios para la programación, a la hora de diseñar eficazmente un software orientado a objetos. Los patrones GRASP describen entre otros elementos la asignación de responsabilidades a objetos, expresados en forma de patrones. Entre los patrones GRASP utilizados se encuentran(81), (82):

Controlador: es un patrón que sirve como intermediario entre una interfaz y el algoritmo que la implementa, de tal forma es la que recibe los datos del usuario y los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Esto se

evidencia en la clase Controladora en la cual se encuentran los métodos que se encargan de la lógica del negocio y esta a su vez pide información a la clase Modelo y envía datos a la Vista_Problema.

Alta Cohesión: Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable. Esto se evidencia en la clase Modelo la cual solo se encarga del acceso a los datos, Vista se encarga de crear y validar el formulario de usuario y Controladora la cual se encarga de la lógica del problema.

2.3. Integración

Uno de los retos en la creación de sitios Web de alta calidad funcional es la integración de aplicaciones. A menudo, se necesitan combinar varias aplicaciones para formar una solución única y fácil de utilizar. El problema de intentar conseguir una solución de este tipo es que las aplicaciones que se necesitan combinar pueden encontrarse en distintas plataformas, cada una ejecutando un sistema operativo distinto. Además, es posible que las aplicaciones se hayan creado en diferentes lenguajes de programación.

Los servicios Web proporcionan un modelo simple, flexible y basado en estándares para conectar aplicaciones a través de Internet. Los servicios Web permiten aprovechar la infraestructura existente en Internet, y enlazar aplicaciones, con independencia de las plataformas, lenguajes de programación o modelos de objetos que se hayan utilizado para implementarlas.(83), (84)

El consorcio W3C(85) define los Servicios Web como sistemas de software diseñados para soportar una interacción interoperable máquina a máquina sobre una red. Los Servicios Web suelen ser APIs Web que pueden ser accedidas dentro de una red (principalmente Internet) y son ejecutados en el sistema que los aloja.

Los servicios pueden ser desarrollados como funcionalidades nuevas o exponer funcionalidades de aplicaciones facilitando la integración entre las nuevas soluciones y las existentes. Estos se implementan en función de tres roles: el consumidor (solicitante del servicio), el proveedor y el intermediario. Los consumidores de servicios acceden a estos mediante interfaces que definen un grupo de operaciones por cada uno de ellos. Estas interfaces son contratos que deben estar separados de la implementación de los servicios, ser auto-descriptivos e independientes de la plataforma tecnológica para lograr un correcto intercambio de mensajes. Para la descripción de los servicios se emplea el lenguaje WSDL (Web Services Description Language).

- Representational State Transfer (REST)

REST es un estilo de arquitectura de software para sistemas hipermedias distribuidos tales como la Web. El término fue introducido en la tesis doctoral de Roy Fielding en 2000, quien es uno de los principales autores de la especificación de HTTP. En realidad, REST se refiere estrictamente a una colección de principios para el diseño de arquitecturas en red. Estos principios resumen cómo los recursos son definidos y diseccionados. El término frecuentemente es utilizado en el sentido de describir a cualquier interfaz que transmite datos específicos de un domino sobre HTTP sin una capa adicional, como hace SOAP. Estos dos significados pueden chocar o incluso solaparse. Es posible diseñar un software de gran tamaño de acuerdo con la arquitectura propuesta por Fielding sin utilizar HTTP o sin interactuar con la Web. Así como también, es posible diseñar una simple interfaz XML+HTTP que no sigue los principios REST, y en cambio seguir un modelo RPC (Llamada a Procedimientos remotos, del inglés *Remote Procedure Call*).

Cabe destacar que REST no es un estándar, ya que es tan solo un estilo de arquitectura. Aunque REST no es un estándar, está basado en estándares:

- HTTP
- URL
- Representación de los recursos: XML/HTML/GIF/JPEG
- Tipos MIME: text/xml, text/html

Las ventajas de la aproximación basada en REST recaen en la potencial escalabilidad de este tipo de sistema. (86)

REPREA utiliza REST para la integración con los demás sistemas.

2.3.1. Conexión

REPREA tiene que resolver los problemas de diseño de los REA de dos formas:

- Los errores entrados por el diseñador de REA
- Los errores que están almacenados en RHODA

RHODA es un repositorio desarrollado en la Universidad donde se almacenan REA, el cual contribuye a elevar la cantidad y calidad de los recursos educativos generados en las Instituciones de Educación Superior. (87)

REPREA para realizar la conexión con RHODA utiliza el servicio web REST. REPREA publica el servicio con el método ResolverProblema(String problema):Lista<Patron> que a partir de un problema pasado por parámetro devuelve los patrones para su solución y RHODA podrá consumir este servicio para resolver los errores de diseño que devolvió el sistema SISDREA (Sistema basado en casos para la identificación de problemas en los diseños de los recursos educativos abiertos).

Para realizar la recomendación de los patrones REPREA se conecta a varios repositorios de patrones a partir del consumo del servicio web REST. Por tanto, los repositorios para poder realizar la conexión con ellos tienen que publicar un servicio web REST con un método que devuelva una lista con todos los patrones que hay en almacenados en ese repositorio.

Para el caso de esta investigación se realizó la conexión con el repositorio de CRODA.

CRODA es una herramienta web desarrollada en la Universidad que posibilita a los docentes la creación y edición de recursos educativos reutilizables, accesibles, duraderos e interoperables de manera independiente. (88)

Actualmente, se desarrolló en esta herramienta un módulo de gestión de patrones de diseño que es el resultado de la investigación titulada “Módulo para la gestión de patrones de diseño de recursos educativos abiertos” y esa es la base de casos de los patrones que se utilizan para realizar la recomendación.Los datos que están almacenados son el nombre del patrón, la descripción del problema, la clasificación, el contexto, el contexto resultante, el catálogo al que pertenece y la información adicional. La comparación entre el problema entrado y el almacenado en la base de casos se realiza mediante el campo “Descripción del problema”.

Para entender mejor el funcionamiento de REPREA y la integración con los repositorios se muestra la siguiente figura:

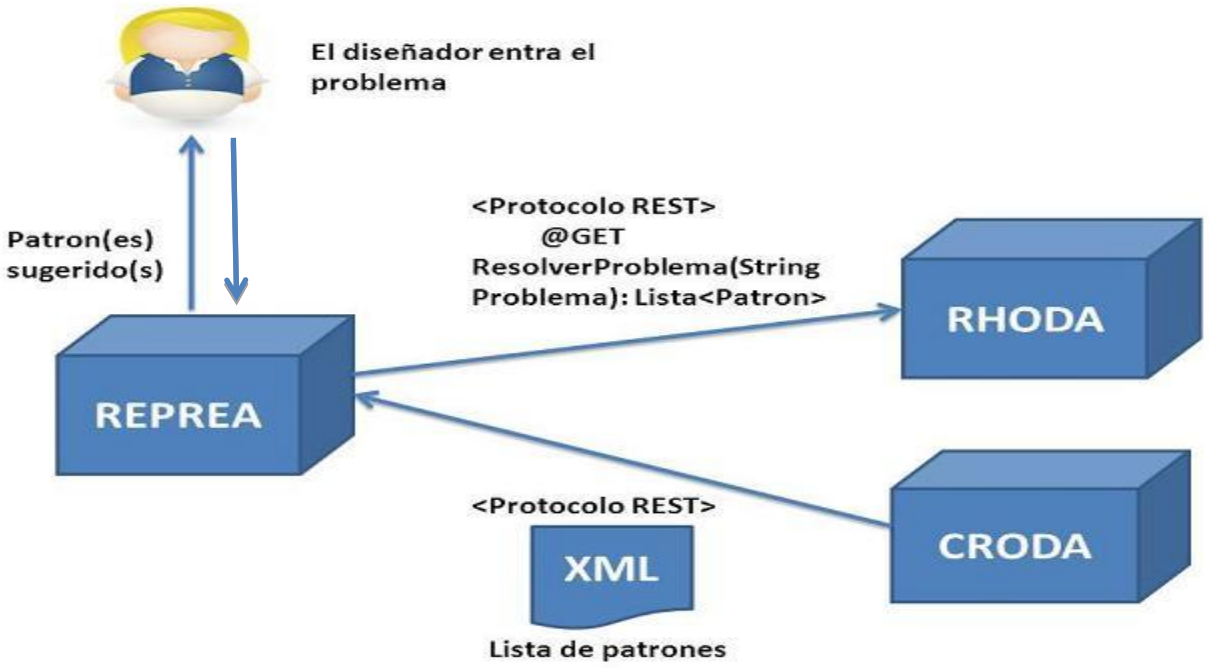


Figura 8.Integración

2.4. Conclusiones parciales

Las fases de recuperación y reutilización de casos permitieron guiar el desarrollo de REPREA. En la etapa de recuperación de casos, la identificación de descriptores relevantes del texto entrado al sistemase logró aplicando la técnica de los n-gramas contextuales. El método de recuperación por analogía utilizado con las medidas de contención y semejanza posibilitó la correcta comparación del problema con los casos almacenados en la base de casos. La selección de los casos que más se ajustan al problema que se desea resolver a partir de los casos recuperados en la fase anterior fue el resultado de la fase de reutilización de casos. El patrón arquitectónico modelo-vista-controlador y el patrón de diseño GRASP utilizados en la implementación de REPREA posibilitaron una organización adecuada del código de la aplicación.

Capítulo 3. Validación del sistema implementado

Introducción

Todos los sistemas presentan errores debido a que son desarrollados por humanos quienes pueden cometer equivocaciones, pero es en la fase de pruebas donde todos los posibles errores cometidos se descubren y se corrigen. En el presente capítulo se valida el resultado obtenido de la investigación y se muestran los resultados de la realización de las pruebas de software la aplicación, en diferentes escenarios.

3.1. Pruebas

El único instrumento adecuado para determinar el status de la calidad de un producto software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. El diseño y ejecución de pruebas tiene como objetivos(89),(90):

1. Encontrar el mayor número de errores con la menor cantidad de tiempo y esfuerzo posibles.
2. Mostrar hasta qué punto las funciones del software operan de acuerdo con las especificaciones y requisitos del cliente.
3. Validar y controlar formalmente la calidad del trabajo realizado.
4. Mejorar la calidad del producto del software.
5. Encontrar y documentar los defectos que puedan afectar la calidad del software.
6. Validar que el software trabaje como fue diseñado.
7. Validar y probar los requisitos que debe cumplir el software.
8. Validar que los requisitos fueron implementados correctamente.

Principios para la aplicación de pruebas

A continuación se exponen diez de los principios de realización de pruebas más relevantes que se tienen en cuenta a la hora de diseñar y aplicar pruebas de software a un sistema. Estos principios son necesarios tenerlos en cuenta en el momento de diseñar los casos de pruebas ya que orientan y guían a los responsables de las pruebas. (91), (92) En el desarrollo de las pruebas a REPREA se aplican estos principios.

1. La prueba puede ser usada para mostrar la presencia de errores, pero nunca de su ausencia.

2. La principal dificultad del proceso de prueba es decidir cuándo parar.
3. Evitar casos de pruebas no planificados a menos que el programa sea verdaderamente sencillo.
4. Una parte necesaria de un caso de prueba es la definición del resultado esperado.
5. Los casos de pruebas tienen que ser escritos no solo para condiciones de entrada válidas y esperadas sino también para condiciones no válidas e inesperadas.
6. Los casos de pruebas tienen que ser escritos para generar las condiciones de salida deseadas.
7. El número de errores sin descubrir es directamente proporcional al número de errores descubiertos.
8. Las pruebas deberían empezar por “lo pequeño” y progresar hacia “lo grande”.
9. Con la excepción de las pruebas de unidad e integración, un programa no deberá ser probado por la persona u organización que lo desarrolló.
10. Las pruebas deberían ser realizadas por un equipo independiente.

Existen dos métodos de pruebas fundamentales: el método de caja negra y el de caja blanca. El método de caja negra consiste en pruebas que se realizan sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software sean operativas. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener en cuenta la estructura interna del software. Además, permite obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. (94), (95)

Por otro lado, en las pruebas de caja blanca se comprueban los caminos lógicos del software proponiendo casos de prueba donde se ejerciten conjuntos específicos de condiciones y/o bucles. Además, se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con lo esperado o mencionado. (94), (95)

En este caso 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 el método de caja blanca al método “similitud entre dos cadenas” para realizar una revisión minuciosa de su código fuente, ya que este es el encargado de realizar el proceso fundamental del sistema, que es generar la recomendación que posteriormente visualizará el usuario. (93), (95)

3.1.1. Pruebas de caja negra

Las pruebas de caja negra, también denominadas, pruebas de comportamiento (Figura 14), se concentran en los requisitos funcionales del software. Es decir, permiten al ingeniero de software derivar conjuntos de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales de un programa. La prueba de caja negra no es una opción frente a las técnicas de caja blanca, sino un enfoque complementario que tiene probabilidades de descubrir una clase diferente de errores de los que descubrirían con los métodos de caja blanca. Este método de prueba es realizado a nivel de sistema, es decir no se tiene ninguna relación con el código del producto. (95), (96)

Figura 9.Prueba de caja negra

A continuación se describen los casos de pruebas desarrollados, especificando la información de entrada, los resultados obtenidos una vez ejecutado el caso de prueba y las condiciones que deben cumplirse mientras este se ejecuta.

Tabla 2. Caso de prueba conexión

Caso de prueba
Nombre: Conexión
Entrada/Pasos de ejecución: El usuario accede a realizar la conexión con el repositorio de patrones. La entrada consiste en introducir el usuario y contraseña, además del ip de la máquina donde se encuentra el repositorio. Pasos de ejecución: En el campo de texto Dirección ip se introduce el ip de la máquina a la cual se va a

conectar, en el campo de texto usuario se introduce el nombre de usuario y en el campo de texto contraseña se introduce la contraseña correspondiente. Una vez introducidos estos datos, se presiona el botón “Conectar”.
Resultado esperado: El sistema muestra un mensaje: Especifique usuario y contraseña en el caso de que haya algún problema con los mismos, en caso contrario el sistema se conecta al repositorio.
Condiciones: La operación se repite hasta que el usuario especifique el usuario y contraseña correctos por el cual va a realizar la conexión.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 3.Caso de prueba recomendación

Caso de prueba
Nombre: Recomendación
Entrada/Pasos de ejecución: El usuario accede a realizar la búsqueda de un patrón de diseño.La entrada consiste en introducir el problema de REA que se quiere resolver. Pasos de ejecución: En el campo de texto problema de REA se introduce el problema que se desea resolver. Una vez introducido, se presiona el botón “Recomendación”.
Resultado esperado: El sistema muestra un mensaje: “Introduzca el problema” en caso de que se oprima el botón “Recomendación” sin haber introducido ningún problema, el sistema muestra un mensaje: “Cambie el problema porque no se encontró ningún patrón semejante” en caso de que en el repositorio no se encuentre ningún patrón que se asemeje al problema introducido y el sistema muestra el formulario con la lista de resultados que coinciden con el criterio de búsqueda en caso contrario.
Condiciones: La operación se repite hasta que el usuario especifique correctamente un problema que tenga

correspondencia con los patrones que están almacenados en el repositorio.
Evaluación de la prueba: Prueba satisfactoria.

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 (97):

1. Significativas: Son aquellas que afectan la calidad del producto o servicio de manera visible, impidiendo o no el cumplimiento de algún requisito.

2. No significativas: Son aquellas que resultan menos visibles, que no atentan el cumplimiento de algún requisito.

3. Recomendaciones: Son aquellas que quedan en función de la apreciación del probador para oportunidades de mejoras del producto o servicio.

Para corregir los errores cometidos durante la implementación de REPREA se realizaron dos iteraciones de pruebas, en las cuales se detectaron un conjunto de no conformidades todas no significativas. En la primera iteración se detectaron dos no conformidades y se resolvieron ambas y en la segunda iteración se detectó una nueva no conformidad a la cual se le dio respuesta, quedando REPREA listo para su uso.

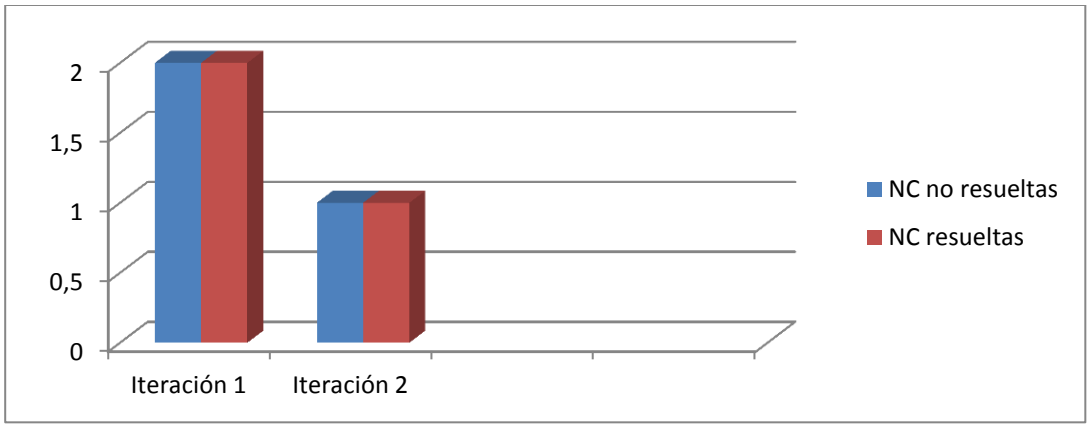


Figura 10.Clasificación de no conformidades

3.1.2. Pruebas de caja blanca

La prueba de caja blanca, en ocasiones llamada prueba de cristal (Figura 13), es un método de diseño que usa la estructura de control descrita como parte del diseño al nivel de componentes para derivar los casos de prueba. Al emplear los métodos de prueba de caja blanca, el ingeniero del software podrá derivar casos de prueba que garanticen que todas las rutas independientes dentro del módulo se han ejercitado por lo menos una vez. Con la aplicación de esta prueba también se puede verificar que se ejecuten todos los bucles dentro de sus límites operacionales y las estructuras de datos internos para asegurar su validez. (95), (98)

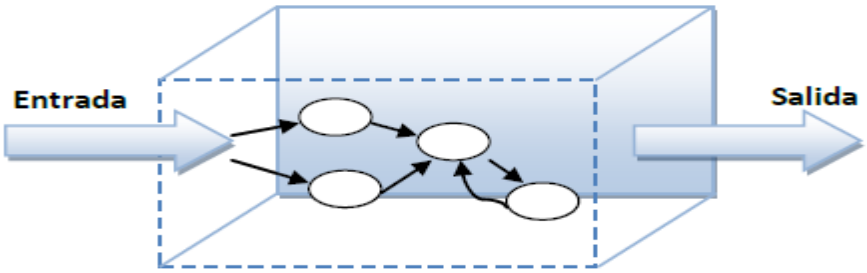
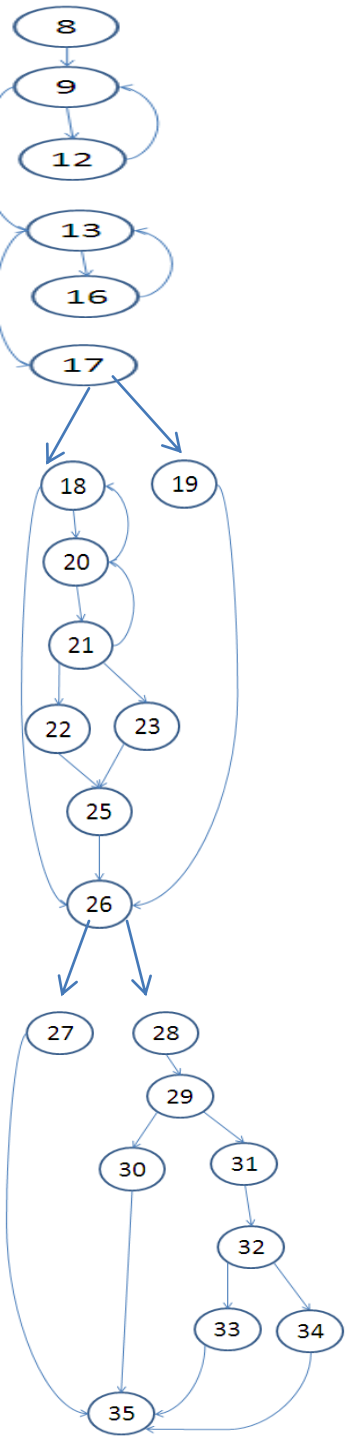


Figura 11.Prueba de caja blanca

En la aplicación de esta prueba se utilizó la técnica de la ruta básica. Esta técnica permite que el diseñador de casos de prueba obtenga una medida de complejidad lógica de un diseño procedimental y que use esta medida como guía para definir un conjunto básico de rutas de ejecución. Los casos de prueba derivados para ejercitar el conjunto básico deben garantizar que se ejecuta cada instrucción del programa por lo menos una vez durante la prueba. (95), (98)

Paso 1



Paso 2

Complejidad ciclomática: Aristas - Nodos + 2 = 32 - 23 + 2 = 11

Paso 3 Caminos independientes que coincide con la complejidad ciclomática

CI1=1->8->9->13->17->19->26->27->35

CI2=1->8->9->12->9->13->17->19->26->27->35

CI3=1->8->9->13->16->13->17->19->26->27->35

CI4=1->8->9->13->17->18->20->21->22->25->26->27->35

CI5=1->8->9->13->17->18->20->18->21->22->25->26->27->35

CI6=1->8->9->13->17->18->26->27-35

CI7=1->8->9->13->17->18->26->28->29->30->35

CI8=1->8->9->13->17->18->20->21->20->21->22->25->26->28->29->31->32->33->35

CI9=1->8->9->13->17->18->20->18->21->20->21->22->25->26->27->35

CI10=1->8->9->13->17->18->20->21->23->25->26->27->35

CI11=1->8->9->13->17->18->20->21->23->25->26->28->29->31->32->34->35

Paso 4

Caso de prueba para: CI1=1->8->9->13->17->19->26->27->35

Entrada: Cadena de texto1, Cadena de texto2.

Resultado esperado: 0 (Significa que las cadenas de texto son iguales).

Resultado de prueba: Satisfactorio.

Caso de prueba para: CI6=1->8->9->13->17->18->26->27-35

Entrada: Cadena de texto1, Cadena de texto2.

Resultado esperado: -1 (Significa que no hay similitud entre las cadenas).

3.2. Técnica de Iadov

En el caso de la presente investigación se utilizó la técnica de Iadov con el fin de conocer el nivel de satisfacción de los diseñadores de REA. Fue necesario formular tres preguntas cerradas en un cuestionario, que se relacionan a través del Cuadro Lógico de Iadov.

La técnica de Iadov en su versión original fue creada por su autor para el estudio de la satisfacción por la profesión en carreras pedagógicas (99). Posteriormente se ha generalizado su uso en otras áreas.

Tabla 4.Técnica de lavod

3- ¿Cuál es la opinión sobre el sistema REPREA desarrollado en la presente investigación?	1- ¿Considera usted factible el desarrollo de un sistema de recomendación para facilitar el proceso de selección de patrones de diseño de REA?								
	No			no sé			Sí		
	2- ¿Si usted necesitara seleccionar un patrón para resolver un problema de diseño de un REA utilizaría REPREA?								
	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.	2	3	6	3	3	3	6	3	4

El número resultante de la interrelación de las tres preguntas indica la posición de cada encuestado en la escala de satisfacción siguiente (99):

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

Si un encuestado responde a la pregunta uno "no", se va a la zona izquierda del cuadro, debajo de la pregunta uno, donde aparece “no”. Si a la pregunta dos responde "no sé" se busca el "no sé" que aparece

debajo del “no” anterior. Si a la pregunta tres responde: "Me disgusta más de lo que me gusta" entonces se busca en las filas, a la izquierda, la casilla donde aparece esa respuesta y se busca el punto donde se interceptan la fila "Me disgusta más de lo que me gusta" con la columna "no sé". El resultado de dicho encuestado es "3", que equivale a "satisfacción no definida". Así se procede con cada usuario de la muestra, en dependencia de sus respuestas. De esta forma se van clasificando en las 6 categorías antes mencionadas.

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en una escala numérica que oscila entre +1 y - 1 de la siguiente forma (99):

Tabla 5.Satisfacción

Escala	Nivel de satisfacción
+1	Máximo de satisfacción
0.5	Más satisfecho que insatisfecho
0	No definido y contradictorio
-0.5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción

La satisfacción grupal se calcula por la siguiente fórmula:

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

En esta fórmula A, B, C, D, E, representan el número de encuestados con índice individual (1; 2; 3; 4; 5 ó 6) y N representa el número total de la muestra.

El índice grupal arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción.

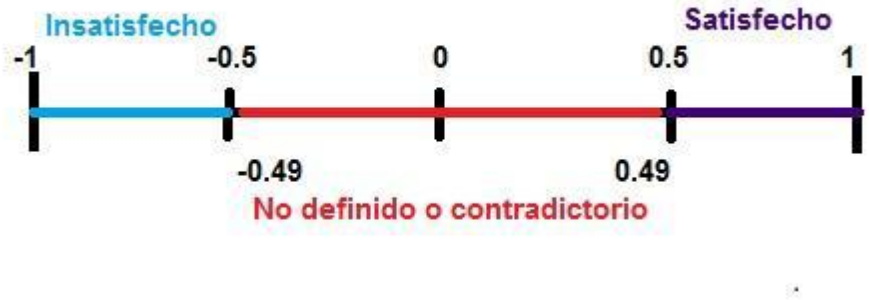


Figura 12.Nivel de satisfacción

Para aplicar la técnica se escogieron como población miembros del proyecto RHODA, se seleccionó una muestra de 8 personas. Después del análisis de los cuestionarios aplicados se obtuvo el siguiente resultado:

Tabla 6.Resultados

Total de usuarios de la muestra (N).	10
Clara satisfacción	6
Más satisfecho que insatisfecho	3
No definida	1
Más insatisfecho que satisfecho	0
Clara insatisfacción	0
Contradictoria	0

A partir de estos resultados para calcular el ISG, las variables de la fórmula tomarían los siguientes valores:

Tabla 7.Valores de la fórmula

Variable	Valor
A	6
B	3
C	1
D	0
E	0

Calculando el ISG quedaría de la forma siguiente:

ISG = $\frac{6 \ +1 \ \ +3 \ +0.5 \ +1 \ 0 \ +0 \ -0.5 \ +0(-1)}{10}$

ISG = $\frac{7.5}{10}$

ISG = 0.75

El ISG resultante está dentro del intervalo de 0,5 a 1, por tanto el nivel de satisfacción de la muestra es: Satisfecho.

3.3. Conclusiones parciales

A partir de la aplicación de las pruebas de caja blanca y de caja negra, realizando las técnicas de ruta básica y partición equivalente, respectivamente, se comprobó el correcto y total funcionamiento de REPREA. Además, el uso de la técnica Iadov, permitió evaluar el nivel de satisfacción de la muestra seleccionada con respecto a REPREA demostrando en los resultados obtenidos que se resolvió el problema planteado en la investigación.

Conclusiones

Con el desarrollo de la presente investigación, se le dio cumplimiento a los objetivos propuestos inicialmente, por lo que una vez analizados los resultados se arriba a las siguientes conclusiones:

- El estudio de los diferentes sistemas de recomendación permitió seleccionar el razonamiento basado en casos textuales para el desarrollo de la aplicación y dentro de su ciclo de vida se utilizaron las fases Recuperación y Reutilización de casos.
- Aplicando la técnica de los n-gramas contextuales en la fase de recuperación de casos fue posible identificar descriptores relevantes del problema entrado al sistema.
- La correcta comparación del problema con los casos almacenados en la base de casos se logró partir del método de recuperación por analogía con las medidas de contención y semejanza.
- A partir de la aplicación de las pruebas de caja blanca y de caja negra se comprobó el correcto y total funcionamiento de REPREA y la técnica de Iadov permitió demostrar que se resolvió el problema planteado en la investigación.

Recomendaciones

- Se recomienda la integración del sistema REPREA con otros repositorios de patrones de diseño de REA.

Referencias bibliográficas

1. **Martínez Quiroz, Max.** La World Wide Web como poderosa herramienta didáctica en la educación a distancia. [En línea] http://lsm.dei.uc.pt/ribie/docfiles/txt200352153013LA%20WORLD%20WIDE%20WEB.pdf.

2. **Zacca González, Grisel y Diego Olite, Francisca.** Los recursos educativos abiertos . [En línea] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-21412010000300008.

3. Revista de Educación a Distancia . [En línea] http://revistas.um.es/index.php/red/article/view/24521.

4. ¿Qué es un Patrón de Diseño? . [En línea] https://msdn.microsoft.com/es-es/library/bb972240.aspx.

5. **Gamma, Erich, y otros, y otros.***Design Patterns (Elements of Reusable Object-Oriented Software).*

6. **Olivares Rojas, Juan Carlos.** Patrones de diseño. [En línea] http://dsc.itmorelia.edu.mx/~jcolivares/courses/dp07b/patrones.pdf.

7. **Ricci, Francesco, Rokach, Lior y Shapira, Bracha.***Recommender Systems Handbook.* 2011.

8. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions . [En línea] 2005. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1423975&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D1423975.

9. **UNESCO.** Forum on the Impact of Open Courseware for Higher Education in Developing Countries. [En línea] 2002. http://unesdoc.unesco.org/images/0012/001285/128515e.pdf.

10. **Atkins, DAniel E, Seely Brown, J y Hammond, Allen L.** A Review of the Open Educational Resources (OER) Movement: Achievements, Challenges, and New Opportunities . [En línea] 2007. http://www.hewlett.org/uploads/files/ReviewoftheOERMovement.pdf.

11. **OECD, Organization for Economic Co-operation and Development:.** Giving Knowledge for Free: The Emergence of Open Educational Resources. [En línea] 2007. http://www.oecd.org/edu/imhe/38947231.pdf.

12. Open Educational Resources infoKit. [En línea] https://openeducationalresources.pbworks.com/w/page/24836860/What%20are%20Open%20Educational%20Resources.

13. Open Educational Resources. [En línea] http://www.hewlett.org/programs/education/open-educational-resources.

14. Eduteka. [En línea] http://www.eduteka.org/OER.php.

15. **OECD, Organization for Economic Co-operation and Development:.** Giving Knowledge for Free: The Emergence of Open Educational Resources. [En línea] 2007. http://www.oecd.org/edu/ceri/38654317.pdf.

16. **Castellanos, Luis.** Hablemos de Recursos Educativos Abiertos. [En línea] https://luiscastellanos.files.wordpress.com/2014/06/recursos-educativos-abiertos.pdf.

17. **UNESCO.** Directrices para los Recursos Educativos Abiertos REA. [En línea] http://www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/CI/CI/pdf/publications/oer_guidelines_es.pdf.

18. **Armengol Garreta, Dani.** Experiencias paralelas: Christopher Alexander y los patrones de diseño. [En línea] junio de 2007. http://www.usolab.com/wl/2007/06/experiencias-paralelas-christo-1.php.

19. **Zapata Ros, Miguel.** Patrones en elearning. Elementos y referencias para la formación. [En línea] 2011. http://www.um.es/ead/red/27/patrones.pdf.

20. **Gónzalez Flores, Simón Carlos.** Diseño Educativo con Patrones de Objetos de Aprendizaje. [En línea] http://ixil.izt.uam.mx/pd/lib/exe/fetch.php/art1tatoaje4to.pdf.

21. **González Aguña, Alexandra.** Patrones en aprendizaje: Concepto, Aplicación y diseño de un patrón. [En línea] http://www.um.es/ead/red/31/alexandra.pdf.

22. **Cáceres Tello, Jesús.***Patrones de diseño: ejemplo de aplicación en los Generative Learning Object.*

23. **Perny, P. y Zucker, J.D.** Sistema de recomendación de restaurantes georreferenciados. [En línea] septiembre de 1999. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.26.4066&rep=rep1&type=pdf.

24. **Xiao, Bo y Benbasat, Izak.**E-commerce product recommendation agents: use, characteristics, and impact. [En línea] 2007. http://dl.acm.org/citation.cfm?id=2017335.

25. **Goldberg, David, y otros, y otros.**Using Collaborative Filtering to weave an information tapestry. [En línea] 1992. https://www.ischool.utexas.edu/~i385d/readings/Goldberg_UsingCollaborative_92.pdf.

26. **Seguido Font, Miguel.** Sistemas de recomendación para webs de información sobre la salud. *Sitio Web UPCOmmons.* [En línea] 2009. http://upcommons.upc.edu/pfc/bitstream/2099.1/7193/6/Master%20Thesis%20Seguido.pdf.txt.

27. **Perugini, Saverio, Goncalves, Marcos Andrés y Fox, Edward A.** A Connection-Centric Survey of Recommender Systems Research. [En línea] http://arxiv.org/pdf/cs/0205059.pdf.

28. **Seguido Font, Miguel.***Sistemas de recomendación para webs de información sobre la salud.* 2009.

29. **Malone, T.** The information lens: an intelligent system for information sharing in organizations. [En línea] https://archive.org/details/informationlens.

30. **Bertate, Leticia, Machado, Rofrigo y Molina, Valeria.** PGMúsica Sistema de Recomendación de música. [En línea] 2006. http://www.fing.edu.uy/inco/grupos/pln/prygrado/InformePGMusica.pdf.

31. **Melville, Prem y Sindhwani, Vikas.**Recommender Systems. [En línea] http://vikas.sindhwani.org/recommender.pdf.

32. **Jannach, Dietmar y Gerhard, Friedrich.**Tutorial: Recommender Systems. [En línea] 2013. http://ijcai13.org/files/tutorial_slides/td3.pdf.

33. **Peña Ayala, Alejandro.** Sistemas basados en Conocimiento:Una Base para su Concepción y Desarrollo. [En línea] http://www.wolnm.org/apa/articulos/Sistemas_Basados_Conocimiento.pdf.

34. **Cingolani, Enrique Antonio.** Evaluación de Sistemas Recomendadores de Contenidos Educativos a través de Estudios de Usuarios. [En línea] agosto de 2014. http://imgbiblio.vaneduc.edu.ar/fulltext/files/TC114681.pdf.

35. **Diz Monje, Cristina, Guzmán Garate, José y Sánchez García, Jesús.** Catalogador automático de textos y Recomendador de artículos del Portal de Revistas Electrónicas de la biblioteca UCM . [En línea] 2008. http://eprints.ucm.es/9887/1/Memoria_Proyecto.pdf.

36. **Lozano, Laura y Fernández, Javier.***Razonamiento Basado en Casos: Una Visión General.*

37. **Castillo Duvergel, Yoannia y Arenas Pérez, Raidel.** Implementación de algoritmos de recomendaciones de imágenes para textos noticiosos. [En línea] 2009. http://catalogoenlinea.uci.cu/cgi-bin/koha/opac-detail.pl?biblionumber=7865.

38. **Ferrera Cedeño, Elvis.** Sistema de recomendación basado en las preferencias de los usuarios para la plataforma PTARTV. [En línea] 2011. http://catalogoenlinea.uci.cu/cgi-bin/koha/opac-detail.pl?biblionumber=10640.

39. **Yera Toledo, Raciel.** Concepción y desarrollo de un sistema de recomendación para jurados online de programación. [En línea] 2010. http://catalogoenlinea.uci.cu/cgi-bin/koha/opac-detail.pl?biblionumber=8643.

40. **Delgado Éxposito, Erly.** Metodologías de desarrollo de software. ¿Cuál es el camino? [En línea] : http://www.monografias.com/trabajos60/metodologias-desarrollo-software/metodologias-desarrollo-software.shtml#ixzz3UfFe0NxP.

41. Recluit atracción del talento en IT. [En línea] http://www.recluit.com/que-lenguajes-de-programacion-dominaran-el-2015/.

42. Unión Informática. [En línea] 2015. http://unioninformatica.org/cual-es-el-lenguaje-de-programacion-mas-usado/.

43. ¡Aprende a programar! [En línea] http://www.taringa.net/posts/apuntes-y-monografias/14033966/Aprende-a-programar.html.

44. Todo sobre PHP. [En línea] 2011. http://www.TODO SOBRE PHP VENTAJAS Y DESVENTAJAS.htm.

45. **Belmonte Fernández, Oscar.** Introducción al lenguaje de programación Java. [En línea] http://www3.uji.es/~belfern/pdidoc/IX26/Documentos/introJava.pdf.

46. History of Java programming language. [En línea] http://www.freejavaguide.com/history.html.

47. **García de Jalón, Javier.***Aprenda Java como si estuviera en primero.* 2000.

48. **Carlos, Jim.** Academica,Comunidad digital del conocimiento. [En línea] http://www.academica.mx/blogs/lenguaje-programaci%C3%B3n-en-java.

49. **Sánchez, Javier.** Base de datos . [En línea] http://es.slideshare.net/gusec/gestion-de-base-de-datos-15058685.

50. **Pecos Martínez, Daniel.** PostGreSQL vs. MySQL. [En línea] http://danielpecos.com/documents/postgresql-vs-mysql/.

51. SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems. [En línea] https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems.

52. **Diz Monje, Cristina, Guzmán Garate, José y Sánchez García, Jesús.** Catalogador automático de textos y recomendador de articulos del portal de revistas electronicas. [En línea] 208. http://eprints.ucm.es/9887/1/Memoria_Proyecto.pdf.

53. **Funk, Peter.**Textual CBR system using domain specific ontology. [En línea] http://www.idt.mdh.se/utbildning/exjobb/files/TR0600.pdf.

54. **Recio, Juan A, y otros, y otros.**Extending jCOLIBRI for Textual CBR. [En línea] http://home.cc.gatech.edu/ccl/uploads/47/jColibiri_TCBR.pdf.

55. **Zarco Fernández, Carmen.** Aplicación de los algoritmos evolutivos al aprendizaje automático de consultas booleanas extendidas para sistemas de recuperacion de informacion difusos. [En línea] 2002.

http://www.cibernetia.com/tesis_es/LINGUISTICA/LINGUISTICA_APLICADA/DOCUMENTACION_AUTOMATIZADA/1. 5..

56. **Méndez Rodríguez, Eva María.***Metadatos y recuperación de información.* 2002.

57. **Méndez, Francisco Javier Martínez.** RECUPERACIÓN DE INFORMACIÓN: MODELOS, SISTEMAS Y EVALUACIÓN. [En línea] http://eprints.rclis.org/16262/1/libro-ri.PDF.

58. **Aamodt, Agnar y Plaza, Enric.**Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. [En línea] http://www.iiia.csic.es/~enric/papers/AICom.pdf.

59. **Kolodner, Janet L.** An Introduction to Case-Based. [En línea] http://alumni.media.mit.edu/~jorkin/generals/papers/Kolodner_case_based_reasoning.pdf.

60. **Rodríguez-Torrejón, Diego A. y Martín-Ramos, José Manuel.***N-gramas de Contexto Cercano para mejorar la Detección de Plagio.*

61. **Rodríguez Torrejón, Diego Antonio y Martín Ramos, José Manuel.***Detección de plagio en documentos. Sistema externo monolingüe de altas prestaciones basado en n-gramas contextuales.* 2010.

62. N-Grams and Corpus Linguistics. [En línea] http://www.cs.columbia.edu/~kathy/NLP/ClassSlides/Class3-ngrams09/ngrams.pdf.

63. **Montes García, Alejandro.***Tecnologías semánticas parala recepción, edición y distribución de video digital.*

64. Tutorialspoint SimpleEasyElearning. [En línea] http://www.tutorialspoint.com/java/java_string_equalsignorecase.htm.

65. Manual Web Tutoriales sobre Programación. [En línea] http://www.manualweb.net/java/funciones-basicas-con-cadenas/.

66. Spanish stemming algorithm. [En línea] http://snowball.tartarus.org/algorithms/spanish/stemmer.html.

67. **Watson, I.** Case-based reasoning is a methodology not a technology. [En línea] https://www.cs.auckland.ac.nz/~ian/papers/cbr/methodology.pdf.

68. **Gálvez Lio, Daniel.***Sistemas Basados en el conocimiento.* 1998.

69. **Kolodner, J. L. y Jona, M. Y.***Case-Based Reasoning: An Overview 1.* 2000.

70. **Barros, Leliane N., y otros, y otros.** Advances in Artificial Intelligence-SBIA 2012. [En línea] 2012. https://books.google.com.cu/books?id=3Yu5BQAAQBAJ&pg=PA194&lpg=PA194&dq=los+n-gramas+contextuales&source=bl&ots=oaHQxDOB2j&sig=Lxi-Ab5B41BP07ypHW-

axoJTegc&hl=es&sa=X&ei=ZnAxVYGjGYjIsAS1hICoBw&ved=0CEIQ6AEwBQ#v=onepage&q=los%20n-gramas%20contextuales&f=fal.

71. **Barrón Cedeño, Luis Alberto.** Detección automática de plagio en texto. [En línea] 2008. https://riunet.upv.es/bitstream/handle/10251/12186/tesisMaster.pdf?sequence=1.

72. **Pérez Suárez, Airel, y otros, y otros.** Algoritmos jerárquicos y no jerárqicos para la construcción de grupos con traslape en contextos dinámicos. [En línea] 2013. http://www.cenatav.co.cu/doc/RTecnicos/RT%20SerieGris_019web.pdf.

73. **Alvares, Miguel Angel.** Desarrollo web. [En línea] 2 de enero de 2014. http://www.desarrolloweb.com/articulos/que-es-mvc.html.

74. patrones arquitectonicos. [En línea] https://ingeniods.wordpress.com/2013/09/16/patrones-arquitectonicos/.

75. patrones arquitectonicos. [En línea] http://www.lsi.us.es/docencia/get.php?id=1130.

76. libros web. [En línea] http://librosweb.es/libro/jobeet_1_4/capitulo_4/la_arquitectura_mvc.html.

77. **Pavón Mestras, Juan.** Estructura de las Aplicaciones Orientadas a Objetos, patrón Modelo-Vista-Controlador (MVC). [En línea] https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf.

78. Reutilización del Software. Patrones de Diseño. [En línea] http://siul02.si.ehu.es/~alfredo/iso/06Patrones.pdf.

79. ¿Qué es un Patrón de Diseño? [En línea] https://msdn.microsoft.com/es-es/library/bb972240.aspx.

80. Design Patterns. [En línea] https://sourcemaking.com/design_patterns.

81. Using Design Patterns with GRASP. [En línea] https://web.cs.dal.ca/~jin/3132/lectures/dp-13.pdf.

82. Diseño dirigido por responsabilidades con los patrones GRASP. [En línea] http://lsi.ugr.es/~mvega/isoo/algpatrones.pdf.

83. **Meza Martínez, Jorge Iván.***Servicios web.*

84. **Navarro Marset, Rafael.***Rest vs Web Service Modelado. Diseño e Implementación de Servicios Web.* 2007.

85. Sobre el W3C. [En línea] 2015. http://www.w3c.es/Consorcio/.

86. **Orellana, Frank.** Trabajo Final - Web Services REST. [En línea] file:///C:/Documents%20and%20Settings/Yaniel/Escritorio/TESIS/servicios%20web/Trabajo%20Final%20-%20Web%20Services%20REST.htm.

87. **González, Roxana Cañizares.** Repositorio de Recursos Educativos para las Instituciones de la Educación Superior. [En línea] 2012. http://bibliodoc.uci.cu/RDigitales/2013/enero/11/TDoc_0018_12_Resumen.pdf.

88. **Martínez Tabasco, Milene Mayra y Rodríguez Campaña, Wilson.** Desarrollo de un módulo para la construcción colaborativa de objetos de aprendizaje en la herramienta de autor CRODA 2.0. [En línea] 2012. http://bibliodoc.uci.cu/RDigitales/2012/diciembre/11/TD_05577_12.pdf.

89. Pruebas de Software. [En línea] http://materias.fi.uba.ar/7548/PruebasSoftware.pdf.

90. Target Testing. [En línea] http://www.targettesting.co.uk/.

91. **Zapata, Javier.** Principios Fundamentales del Proceso de Pruebas. [En línea] 2013. https://pruebasdelsoftware.wordpress.com/2013/01/07/principios-fundamentales-del-proceso-de-pruebas/.

92. **Meyer, Bertrand.** Seven principles of software testing. [En línea] http://se.ethz.ch/~meyer/publications/testing/principles.pdf.

93. **El-Far, Ibrahim K. y Whittaker, James A.** Model-based Software Testing. [En línea] http://testoptimal.com/ref/Model-based%20Software%20Testing.pdf.

94. **Marick, Brian.** New Models for test development. [En línea] http://www.exampler.com/testing-com/writings/new-models.pdf.

95. Tipos de pruebas de software. [En línea] http://es.slideshare.net/GuillermoLemus/tipos-de-pruebas-de-software.

96. Testing Overview and Black-Box Testing Techniques . [En línea] http://agile.csc.ncsu.edu/SEMaterials/BlackBox.pdf.

97. Intranet SGC. [En línea] https://sites.google.com/a/iesprimerodemayo.com/sgc/preguntas-as-frecuentes/%C2%BFqueesunanoconformidad.

98. White Box Testing. [En línea] http://agile.csc.ncsu.edu/SEMaterials/WhiteBox.pdf.

99. **López Rodríguez, Alejandro y González Maura, Viviana.** La técnica de Iadov. Una aplicación para el estudio de la satisfacción de los alumnos por las clases de educación física . [En línea] http://www.efdeportes.com/efd47/iadov.htm.

Anexos

Anexo 1.Ejemplos de patrones

Especificación del patrón de diseño

Nombre: OA adaptativo al contexto de aprendizaje

Categoría: Diseño de aprendizaje, creación de objetos de aprendizaje reutilizables

Resumen: Una unidad o módulo de contenidos puede ser similar en determinados dominios de conocimiento y por tanto se pueden utilizar recursos de aprendizaje comunes. A pesar de ello, en cualquier dominio de aprendizaje puede resultar muy interesante, en términos pedagógicos, incorporar detalles contextuales en forma de ejemplos, o ilustraciones particulares a ese dominio.

Análisis:

1. Los costes y tiempos de desarrollo de materiales educativos son muy altos.
2. Definir procesos y metodologías aplicadas al diseño de materiales de aprendizaje proporciona estandarización y mayor calidad tanto a nivel organizacional como a nivel del propio proceso de aprendizaje.
3. Excesiva duplicidad de contenidos de aprendizaje y falta de reutilización. Muchos contenidos se basan en los mismos conceptos pero se presentan desde diferentes perspectivas.
4. Analizar las tecnologías implicadas en el desarrollo de los objetos de aprendizaje, valorar alternativas de herramientas software para la creación de objetos de aprendizaje.

Problema: La reutilización de recursos de aprendizaje en diferentes contextos es una cuestión muy importante actualmente en el área del e-learning. En la medida en que se diseña un objeto de aprendizaje de tal forma que puede utilizarse en múltiples contextos se garantiza también la independencia del propio recurso con respecto a dichos contextos. Por otro lado, en cualquier contexto particular, puede resultar útil desde un punto de vista pedagógico incluir elementos en el material didáctico dependientes del contexto como ejemplos gráficos para entender mejor los conceptos, animaciones. Finalmente, hay que mantener un equilibrio entre el hecho de garantizar la reutilización de los materiales didácticos logrando una independencia del contexto y personalizar los contenidos de tal forma que se adecuen a las necesidades de los alumnos y se puedan aplicar a contextos particulares de aprendizaje.

Solución: Como solución se propone diseñar la parte principal del recurso didáctico de forma que sea completamente independiente del contexto pero identificando aquellas partes del mismo dónde se pueden incluir ejemplos o ilustraciones particulares a un contexto de aprendizaje determinado. Incluyendo en el

material la posibilidad de integrar elementos o recursos externos permitimos así que el recurso de aprendizaje se pueda adaptar para su utilización en diferentes contextos. El recurso final y objeto de aprendizaje consistiría en un componente principal, independiente del contexto, que permite la inclusión de elementos externos en determinadas partes proporcionando ejemplos, animaciones.

Ejemplo de uso: En muchos recursos relacionados con el área de programación computacional, la mayoría de conceptos son los mismos independientemente de la especialidad o tipo de estudios. También hay que tener en cuenta la independencia del lenguaje de programación a utilizar ya que una determinada aplicación presenta las mismas funcionalidades pero puede implementarse en distintos lenguajes. Un recurso didáctico que muestra o enseña conceptos de programación es independiente del lenguaje de implementación utilizado, de tal forma que sería muy sencillo desarrollar un cuerpo principal mostrando los conceptos generales y posteriormente añadir ejemplos, o recursos externos en base a diferentes lenguajes de programación en las partes apropiadas. En la figura 12 se presenta un ejemplo de un objeto de aprendizaje muy sencillo basado este patrón.

Otros patrones: GLO CETL “Aprendizaje activo mediante ejemplos visuales”. Análisis de las fases de un proceso de aprendizaje proporcionando elementos o pequeñas plantillas para poder personalizar los contenidos en función del contexto de aprendizaje y necesidades docentes.

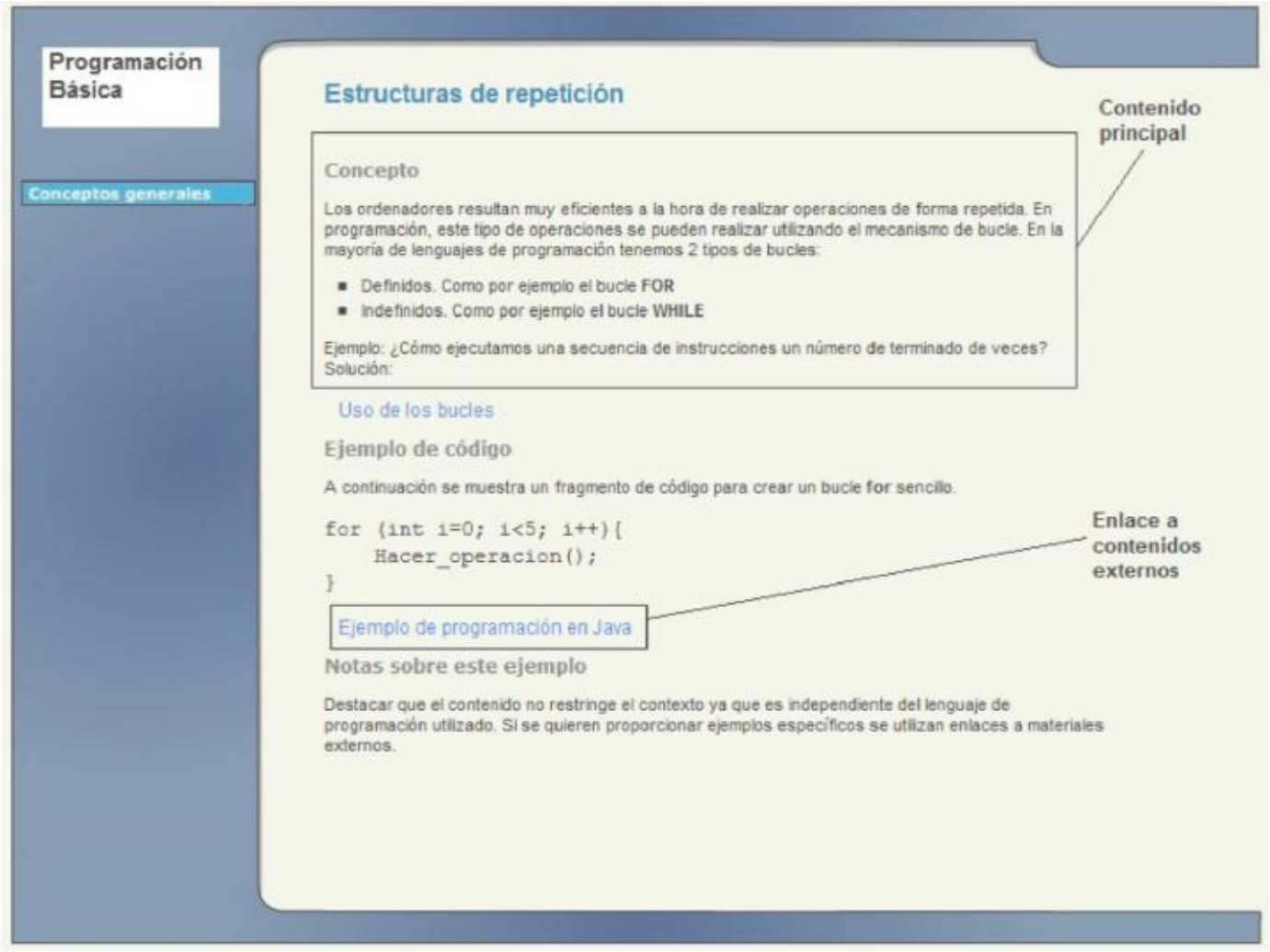


Figura 12. Ejemplo de objeto de aprendizaje basado en el patrón

Los contenidos del objeto de aprendizaje de ejemplo son neutrales con respecto al lenguaje de programación a utilizar. Los enlaces a materiales externos pueden ser independientes del lenguaje o referentes a lenguaje de programación en particular. Si se quisiera utilizar otros lenguajes simplemente habría que cambiar los enlaces a contenidos externos teniendo una nueva versión de forma rápida y sencilla.

Otro ejemplo de patrón pedagógico EVA es el que propone José María Rodríguez: (101)

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.

Un ejemplo de patrón de diseño para crear una mulitimedia para enseñar un concepto de programación es el siguiente: (102)

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.