

Universidad de las Ciencias Informáticas

Facultad 4



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

Sistema basado en casos para la detección de problemas en el diseño
de los recursos educativos digitales versión 2.

Autor: Adriana Valdés Avilés

Tutores: Ing. Yasirys Terry González

Ing. Leduan B. Rosell Acosta

Ing. Reiman Alfonso Azcuy



“En tiempos de cambio, quienes estén abiertos al aprendizaje se adueñarán del futuro, mientras que aquellos que creen saberlo todo estarán bien equipados para un mundo que ya no existe.”

Eric Hoffer

Agradecimientos

A mis padres por haber sido siempre mi apoyo, mis guías en la vida, por entenderme y ayudarme en todos los momentos que lo he necesitado. A mi mamita por ser mi amiga, por sus consejos, su cariño, por siempre pensar en mí, por ser esa madre que todos merecen tener. A mi papito por aceptar mis decisiones y mis sueños y ayudarme a realizarlos, por todos los sacrificios que ha hecho por mí, por su cariño y su comprensión, porque sin él este recorrido no habría sido posible y a mi segunda mamá Chiqui por sus consejos y apoyo incondicional, gracias.

A mi hermanito que no pudo estar aquí pero ahora mismo me debe estar viendo gracias a las tecnologías, muchas gracias.

A mi novio, mi amigo, mi tutor, mi apoyo en todo momento, mi compañero de toda la vida Raciél, le agradezco por su paciencia y cooperación para que esta tesis sea hoy un hecho.

A mis amigos de la UCI por hacer estos 5 años inolvidables, a mis compañeras de apartamento Betty y Yennifer por el cariño y el apoyo.

A mis amigos y compañeros (Victor, Osmani), a los que están y a los que no. En fin, a todos los que de una forma u otra han formado parte de mi vida durante el transcurso de estos años.

A mis tutores por la atención prestada.

Declaración de autoría

Declaro ser la autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Adriana Valdés Avilés

Firma del Autor

Leduan Bárbaro Rosell Acosta

Firma del Tutor

Yasirys Terry González

Firma del Tutor

Reiman Alfonzo Azcuy

Firma del Tutor

Resumen

Por el creciente uso de las tecnologías con el objetivo de apoyar el proceso de enseñanza-aprendizaje se ha desarrollado una gran cantidad de recursos educativos digitales, sin embargo, no siempre tienen la calidad requerida. Para corregir este problema, en la Universidad de las Ciencias Informáticas se desarrolló el sistema SISDREA, el cual, permite determinar cuáles son los posibles problemas de diseño que presenta un recurso educativo, facilitando así, el trabajo para los diseñadores de estos recursos. Como resultado del análisis realizado a SISDREA v1.0 se detectaron debilidades que pudieran resolverse desarrollándose una segunda versión, donde se reutilizó el módulo agente inteligente, al cual se le realizaron cambios en las funciones de semejanza con el objetivo de obtener buenos resultados de precisión. Se agregaron los procesos encargados de realizar la evaluación y el almacenamiento de la solución pertenecientes al ciclo del Razonamiento Basado en Casos. Se incluyeron los módulos gestión de los repositorios y explicación, garantizando la conexión con varios repositorios a la vez y un mejor entendimiento del proceso llevado a cabo para la detección de problemas de diseño en recursos educativos digitales, respectivamente. Como guía al desarrollo de software se empleó la metodología Proceso Unificado Ágil en su variante para la universidad, así como un conjunto de lenguajes y herramientas que cimentan la base tecnológica. Además, para la validación de los resultados obtenidos, se realizaron las pruebas de calidad permitiendo detectar las no conformidades en los componentes, las cuales, fueron corregidas en el menor tiempo garantizando la calidad de las funcionalidades establecidas por el cliente.

Palabras clave: diseño, problema, repositorio, recurso educativo, sistema basado en casos.

Índice

Introducción	1
Capítulo I: Fundamentación teórica de la investigación	5
1.1. Recursos educativos.....	5
1.2. Revisión de recursos educativos digitales.....	7
1.3. Sistemas Basados en Casos.....	8
1.3.1. Ventajas de un sistema basado en casos (11).....	9
1.4. Análisis de SISDREA v1.0.	10
1.4.1. Proceso de recuperación en SISDREA v1.0.....	10
1.4.2. Proceso de adaptación en SISDREA v1.0	12
1.4.3. Resultados del análisis de SISDREA v1.0	13
1.5. Algoritmos de recuperación de casos	15
1.5.1. Medidas de similitud.....	16
1.6. Repositorios de recursos educativos digitales.....	17
1.7. Interoperabilidad	18
1.8. Lenguajes, herramientas y metodología a utilizar	19
1.8.1. Metodología de Software.....	20
1.9. Soluciones existentes a nivel internacional y nacional.....	25
Conclusiones parciales del capítulo	29
Capítulo II: Propuesta de solución para SISDREA v2.0	30
Introducción.....	30
2.1. Descripción del sistema.....	30
2.1.1. Módulo de agente inteligente.....	31

2.1.2. Módulo de conexión con los repositorios	34
2.1.3. Módulo de explicación	35
2.1.4. Similitudes y diferencias entre SISDREA v1.0 y SISDREA v2.0	35
2.2. Especificación de requisitos del software	37
Requisitos no funcionales	39
2.3. Historias de usuario (HU)	41
2.4. Arquitectura de software	42
2.5. Patrones de diseño:	43
2.5.1. Patrones GRASP	43
2.5.2. Patrones GoF	45
2.6. Diagrama de clases del diseño	46
2.7. Diagrama de paquetes.....	48
Conclusiones parciales del capítulo	49
Capítulo III: Implementación y prueba de SISDREA v2.0.....	50
Introducción	50
3.1. Estándar de codificación	50
3.2. Pruebas de software	51
3.2.1. Pruebas unitarias.....	52
3.2.2. Pruebas de sistema.....	61
3.2.3. Pruebas de regresión.....	64
Conclusiones.....	66
Recomendaciones.....	67
Referencias bibliográficas	68

Índice de tablas

Tabla 1: Principales elementos de un recurso educativo (7), (8).....	5
Tabla 2: Estructura de la base de casos del sistema (elaboración propia).....	10
Tabla 3: Elementos que se mantienen y sufrirán cambios en base al sistema anterior (elaboración propia)	14
Tabla 4: Comparación de metodologías (elaboración propia)	23
Tabla 5: Cuadro comparativo de sistemas basados en casos (elaboración propia).....	27
Tabla 6: Metadatos definidos por LOM que se utilizan en SISDREA (elaboración propia)	31
Tabla 7: Nivel de similitud de la categoría dificultad (elaboración propia).....	33
Tabla 8: Similitudes y diferencias entre SISDREA v1.0 y SISDREA v2.0 (elaboración propia).....	36
Tabla 9: Descripción de los requisitos funcionales (elaboración propia).....	37
Tabla 10: Numeración de los nodos en cada porción del código, método eliminar (elaboración propia)	53
Tabla 11: Numeración de los nodos en cada porción del código, método insertar (elaboración propia)	54
Tabla 12: Fórmulas y resultados por métodos de la complejidad ciclomática (elaboración propia)	56
Tabla 13: Tabla de rutas básicas (elaboración propia).....	56
Tabla 14: Caso de prueba eliminar: Ruta independiente #1 (elaboración propia)	56
Tabla 15: Caso de prueba eliminar: Ruta independiente #2 (elaboración propia)	57
Tabla 16: Caso de prueba eliminar: Ruta independiente #3 (elaboración propia)	57
Tabla 17: Caso de prueba eliminar: Ruta independiente #4 y #5 (elaboración propia)	58
Tabla 18: Caso de prueba insertar repositorio: Ruta independiente #1 (elaboración propia) ...	58
Tabla 19: Caso de prueba insertar repositorio: Ruta independiente #2 (elaboración propia) ...	58
Tabla 20: Caso de prueba insertar repositorio: Ruta independiente #3 (elaboración propia) ...	59
Tabla 21: Caso de prueba insertar repositorio (elaboración propia).....	61
Tabla 22: Nivel de similitud de la categoría estructura (elaboración propia)	75
Tabla 23: Nivel de similitud de la categoría nivel de agregación (elaboración propia)	76
Tabla 24: Nivel de similitud de la categoría estado (elaboración propia).....	76
Tabla 25: Nivel de similitud de la categoría tipo de interacción (elaboración propia)	76
Tabla 26: Nivel de similitud de la categoría tipo de recurso educativo (elaboración propia)	77
Tabla 27: Nivel de similitud de las categorías nivel de interacción y densidad semántica (elaboración propia).....	78
Tabla 28: Nivel de similitud de la categoría propósito (elaboración propia)	78

Índice de figuras

Figura 1: Ciclo básico de un sistema basado en casos (18).....	9
Figura 2: Fórmula para determinar la distancia Euclídea (20).....	16
Figura 3: Módulos que componen la solución propuesta (elaboración propia).....	31
Figura 4: Funciones de similitud para la categoría dificultad (elaboración propia).....	34
Figura 5: Arquitectura en 3 niveles (elaboración propia).....	43
Figura 6: Utilización del patrón Experto en la aplicación (elaboración propia).....	44
Figura 7: Utilización del patrón Creador en la aplicación (elaboración propia).....	44
Figura 8: Utilización del patrón Controlador en la aplicación (elaboración propia).....	45
Figura 9: Relación entre las clases del módulo de recuperación (5).....	46
Figura 10: Relación entre las clases de la biblioteca LomStandarInterpreter (5).....	47
Figura 11: Diagrama de clases módulo de conexión con los repositorios (elaboración propia)	47
Figura 12: Diagrama de clases módulo listar repositorios (elaboración propia).....	48
Figura 13: Diagrama de clases módulo de explicación (elaboración propia).....	48
Figura 14: Diagrama de paquetes de SISDREA v2.0 (elaboración propia).....	49
Figura 15: Utilización del estándar de codificación Upper Camel Case (elaboración propia)	51
Figura 16: Utilización del estándar de codificación Lower Camel Case en la aplicación (elaboración propia).....	51
Figura 19: Grafo de flujo función eliminar repositorio.....	55
Figura 20: Grafo de flujo, función insertar repositorio.....	55

Introducción

Las tecnologías de la información y las comunicaciones son todo lo relativo a la informática conectada a Internet, los medios de comunicación y especialmente el impacto social del uso de estos. Son un conjunto de herramientas electrónicas utilizadas para la recolección, almacenamiento, tratamiento, difusión y transmisión de la información representada de forma variada (1).

El uso de las tecnologías de la información y las comunicaciones se ha convertido en un pilar importante en el ámbito educativo, posibilitando que el aprendizaje sea interactivo y creativo. Para lograr esto, se hace necesario que los recursos educativos digitales tengan la calidad necesaria para ser utilizados. Es por ello que surgen los sistemas de revisión de recursos educativos, los cuales se encargan de detectar problemas de diseño, para que los desarrolladores de los mismos puedan corregirlos y obtener un sistema con la calidad requerida.

Los materiales digitales se denominan Recursos Educativos Digitales (RED) cuando su diseño tiene una intencionalidad educativa, apuntando al logro de un objetivo de aprendizaje. Están hechos para: informar sobre un tema, ayudar en la adquisición de un conocimiento, reforzar un aprendizaje, remediar una situación desfavorable, favorecer el desarrollo de una determinada competencia y evaluar conocimientos (2). Estos objetos pueden ser aprovechados a partir del planteamiento de objetivos educativos y con la inclusión de metadatos que permiten su descripción, lo cual facilita su reutilización y adaptación a diferentes ambientes (3). Entre los tipos de recursos educativos digitales pueden mencionarse: mapas curriculares, materiales de cursos, libros de estudio, videos, aplicaciones, multimedia o cualquier material que haya sido diseñado para la enseñanza y el aprendizaje (4).

La utilización de Recursos Educativos (RE) en los procesos de enseñanza-aprendizaje es una práctica que cada vez ha tomado mayor fuerza. Debido a la creciente demanda de tener a disposición cantidades considerables de RE, que tengan niveles de calidad aceptables para ser utilizados, muchos profesionales e instituciones se han dado a la tarea de especializarse en el proceso de producción de objetos digitales para el apoyo del proceso enseñanza-aprendizaje, sin embargo, aún se manifiestan muchos problemas en sus diseños (5).

Estos problemas de diseño pueden estar relacionados con elementos como: la calidad de los contenidos, adecuación de los objetos de aprendizaje, adaptabilidad, motivación, diseño y presentación, accesibilidad, entre otros.

En los últimos años se ha creado un número importante de herramientas informáticas que apoyan la gestión de recursos educativos. Algunas de ellas incluyen reportes acerca de cuáles de estos presentan diseños defectuosos, como es el caso de RHODA¹, repositorio para el almacenamiento de RE desarrollado en la Universidad de las Ciencias Informáticas (UCI). Sin embargo, en ocasiones a los diseñadores de estos recursos les resulta difícil detectar cuáles son los problemas de diseño específicos, dificultándose su corrección y la creación de patrones que eviten la reiteración de estos problemas. Este impedimento está dado por la complejidad implícita en el proceso de revisión de estos recursos.

En el curso 2014-2015 se desarrolló en la UCI, el sistema SISDREA (5), el cual es un sistema experto, que resuelve la situación problemática anteriormente planteada, sin embargo, presenta aspectos funcionales que pudieran mejorarse para garantizar la comprensión por parte de los usuarios y la obtención de buenos resultados de precisión en las soluciones, así como la interoperabilidad con repositorios donde se almacenan recursos educativos y sus metadatos y con otros sistemas. Los aspectos fundamentales identificados son:

- Las funciones de semejanza que se utilizan para la comparación de los rasgos nominales solo evalúan con valor 1 en caso de ser iguales y 0 en cualquier otro caso.
- No presenta una solución donde se le brinde al usuario información sobre los elementos que se tienen en cuenta para la detección de los problemas de diseño en recursos educativos.
- El sistema presenta limitaciones técnicas para garantizar el intercambio de información con herramientas informáticas que se utilizan con propósitos de apoyar la gestión de recursos educativos y las opciones que brinda en este sentido, se configuran desde el código de la aplicación.
- El sistema presenta carencias relacionadas con el almacenamiento de nuevos casos, que pudieran constituir bases para posteriores análisis.

A partir de la situación problemática descrita anteriormente se plantea como **problema a resolver** ¿Cómo incrementar la comprensión de los resultados brindados por SISDREA y obtener buenos niveles de precisión para la detección de problemas de diseño en recursos educativos?

¹ RHODA Repositorio de Objetos de Aprendizaje.

Teniendo en cuenta lo anteriormente planteado, se define el **objeto de estudio**: proceso de revisión de los recursos educativos; centrándose en el **campo de acción**: razonamiento basado en casos para la revisión de los recursos educativos.

Como **objetivo general** de este trabajo de diploma se establece desarrollar la segunda versión del sistema basado en casos para la detección de problemas en el diseño de recursos educativos digitales, para incrementar la comprensión y ofrecer buenos niveles de precisión en los resultados.

Para dar cumplimiento al objetivo general se establecen los siguientes objetivos específicos:

1. Comprender los basamentos teóricos de SISDREA en su versión 1.0.
2. Elaborar una fundamentación teórica mediante el estudio de la bibliografía relativa a los sistemas basados en casos.
3. Implementar una segunda versión del sistema SISDREA para la identificación de problemas de diseño de recursos educativos digitales.
4. Realizar pruebas al sistema para comprobar el correcto funcionamiento de las funcionalidades implementadas.

Entre los resultados que se esperan con el desarrollo de la presente investigación, se encuentra un sistema que garantice la comprensión de los resultados y la obtención de buenos niveles de precisión en la detección de problemas de diseño en recursos educativos digitales.

Para el desarrollo de la investigación se utilizaron los métodos listados a continuación.

Métodos Teóricos

Análisis histórico-lógico: se utiliza para realizar la valoración acerca de la evolución de las tecnologías vinculadas a los sistemas expertos existentes en la actualidad, potenciando el conocimiento sobre su funcionamiento y estructura.

Analítico-sintético: se utiliza en el estudio de las tecnologías analizando el funcionamiento de las mismas y la relación existente entre ellas para el desarrollo de la investigación.

Métodos Empíricos

Entrevistas: dentro de la entrevista es empleada la semi-estructurada, para la obtención de los requisitos necesarios para satisfacer las expectativas de los clientes.

Consultas bibliográficas: Permite la elaboración del marco teórico de la investigación fundamentada por la información consultada.

El documento está estructurado en tres capítulos, siendo estos:

Capítulo I: Fundamentación teórica de la investigación.

En este capítulo se fundamenta el marco teórico de la investigación, se hace un análisis de la primera versión de SISDREA, definiéndose conceptos importantes para el entendimiento del sistema, además, se estudian sistemas similares tanto a nivel nacional como internacional, se caracteriza la metodología de desarrollo, las herramientas, y lenguaje de programación utilizado en el proceso de desarrollo del sistema basado en caso.

Capítulo II: Propuesta de solución para SISDREA v2.0.

En este capítulo se exponen las características de la propuesta de solución, se detallan los diagramas utilizados para el desarrollo de la solución y se definen los requisitos funcionales y no funcionales necesarios para dar cumplimiento al objetivo general de la investigación.

Capítulo III: Implementación y prueba de SISDREA v2.0.

En este capítulo se describen los elementos de implementación del sistema basado en casos para la detección de problemas de diseño en recursos educativos, se describen los estándares de programación y codificación utilizados y se describen las pruebas realizadas para comprobar el correcto funcionamiento del sistema.

Capítulo I: Fundamentación teórica de la investigación

Capítulo I: Fundamentación teórica de la investigación

Este trabajo se apoya, fundamentalmente, en definiciones teóricas, de diferentes áreas relacionadas con el tema de investigación. Es por este motivo que, resulta necesario dedicar un capítulo al estudio del arte, que comienza con el análisis de la versión anterior del sistema para detección de problemas de diseño de recursos educativos. También incluye el estudio de investigaciones realizadas en las siguientes temáticas: recursos educativos, metadatos, repositorios digitales y sistema basado en casos; describiendo tendencias, buenas prácticas, tecnologías y herramientas que pueden servir como referencia para la construcción de la solución.

Las tecnologías de la información y las comunicaciones encuentran su papel como una especialización dentro del ámbito de la didáctica y de otras ciencias aplicadas de la educación, refiriéndose especialmente al diseño, desarrollo y aplicación de recursos en procesos educativos, no únicamente en los procesos instructivos, sino también en aspectos relacionados con la educación social y otros campos educativos. Estos recursos se refieren, en general, a los recursos de carácter informático, audiovisual, tecnológicos, el tratamiento de la información y los que facilitan la comunicación (6).

A continuación, se muestran las principales características de un recurso educativo, para comprender por qué contribuyen a mejorar el proceso de enseñanza-aprendizaje, así como, los problemas que pudiera presentar un recurso educativo digital.

1.1. Recursos educativos

Los Recursos Educativos Digitales son recursos ofrecidos en línea para docentes y estudiantes; incluyendo contenido educativo muy variado (texto, imágenes, recursos, audio, video, juegos educativos, portales, entre otros) y herramientas de software. Estos recursos se pueden utilizar, compartir, combinar y adaptar según las necesidades educativas. Las principales características que se deben garantizar para que un RED cumpla con el propósito con el que fue creado se resumen en la Tabla 1.

Tabla 1: Principales elementos de un recurso educativo (7), (8).

Adecuación	Se debe determinar de antemano el tramo educativo y el área donde se utilizará el recurso.
Idoneidad	Se elegirán los elementos del currículum que se trabajan mejor.

Capítulo I: Fundamentación teórica de la investigación

Prioridad	En un principio no es posible implementar recursos sobre todos y cada uno de los contenidos del currículum. Por ese motivo se hace necesario establecer una prioridad y ésta puede ser la de las enseñanzas mínimas exigibles.
Necesidad	El diseño de un recurso también puede surgir de una necesidad vivenciada. La falta de motivación de algunos alumnos/as hacia determinados contenidos o la necesidad de hacer algo distinto son justificaciones legítimas para buscar un recurso multimedia, centrándose de antemano en ciertos contenidos del currículum.
Interactividad	El contenido elegido debe permitir al alumno tomar decisiones y apreciar las consecuencias de las mismas. Si se pretende elaborar un “libro digital” con un nivel de interacción mínimo (pasar página) seguramente se sustituirá rápidamente por el libro en papel más inmediato y manejable.
Transferencia	Los contenidos que se trabajen en una multimedia deben tener múltiples referencias a actividades de la vida cotidiana para asegurar una mayor transferencia.
Adaptabilidad y reusabilidad	El diseño de recursos fácilmente personalizables por parte del profesorado permite la adaptación y reutilización en distintas situaciones. Así, por ejemplo, un cuestionario de preguntas donde sea posible modificar fácilmente las preguntas y respuestas es más reutilizable que un cuestionario cerrado. La reusabilidad es la capacidad para usarse en distintos escenarios de aprendizaje y con alumno/as de distintos bagajes.

En relación con estas características, a continuación, se mencionan algunas limitaciones que hacen que los recursos educativos no cumplan con sus objetivos (6).

- El contenido es incorrecto. El contenido presenta omisiones o sesgos.
- El nivel de detalle es inadecuado. Las presentaciones no refuerzan los puntos clave ni las ideas significativas.
- El contenido no es relevante en la vida del alumnado.
- Las actividades son demasiado fáciles o demasiado difíciles para percibir el interés del alumnado.
- Las características del objeto que suponen captar el interés del alumno son una distracción que interfiere con el aprendizaje.

Capítulo I: Fundamentación teórica de la investigación

- El objeto no varía en el tono, no hay muestras de humor o novedad.
- La interacción del alumno/a con el objeto no presenta consecuencias interesantes.
- La fuente seleccionada o el tamaño de la misma reduce notablemente la velocidad de lectura.
- Cierta información que es necesaria para la comprensión del objeto de aprendizaje es ilegible.
- La calidad del vídeo o audio es insuficiente para el aprendizaje.
- La selección de colores, imágenes o sonidos interfieren con los objetivos de aprendizaje.
- El diseño de la información provoca un sobreesfuerzo innecesario para el procesamiento de la misma.
- No hay suficientes encabezados o no son significativos para el alumno/a.
- No hay posibilidad de interactividad, solo hay texto y/o imágenes estáticas.
- Varios enlaces o botones no funcionan.
- Hay una demora excesiva en la navegación.
- El funcionamiento de la interfaz no es intuitivo y tampoco hay instrucciones.
- El funcionamiento de la interfaz es inconsistente e impredecible.
- Los videos no tienen subtítulos.
- Faltan etiquetas en las imágenes.
- La comprensión de los gráficos requiere que el alumno/a pueda percibir los colores.

Para darle solución a los problemas anteriores es necesario desarrollar un sistema que permita realizar la revisión de recursos educativos. A continuación, se especifican los elementos fundamentales de la revisión de recursos educativos digitales.

1.2. Revisión de recursos educativos digitales

Las revisiones son consideradas procesos que se realizan de forma minuciosa y exhaustiva. Según Bobenrieth Astete, cada publicación debe ser evaluada críticamente, con el fin de determinar su veracidad, utilidad, puesta en práctica, importancia para la sociedad y los problemas que la aquejan (7). Las comprobaciones, verificaciones y evaluaciones son actividades o acciones que se realizan en cualquier contexto de la sociedad; lo importante es llevar a cabo un proceso correcto, capaz de cumplir y garantizar el éxito o la calidad requerida (8).

En el entorno educativo es de suma importancia la realización de los procesos de revisión, principalmente cuando se lleva a cabo a través de medios electrónicos e informatizados. Los sistemas de gestión del aprendizaje permiten la realización de procesos de formación e instrucción. La publicación de recursos y la carencia de algún mecanismo de comprobación da margen a la existencia

Capítulo I: Fundamentación teórica de la investigación

de errores de cualquier tipo, logrando así desconfianza y poca motivación a la hora de hacer uso de dichos componentes formativos. Para darle solución a este problema es necesario desarrollar un sistema que permita revisar recursos educativos y proponer posibles problemas de diseño antes de ser publicado o utilizado.

Las características del proceso de revisión de recursos educativos han motivado el uso de sistemas informáticos en este campo, como por ejemplo el repositorio RHODA (9). La primera versión del Sistema basado en casos para la detección de problemas de diseño en recursos educativos utiliza, específicamente, el Razonamiento Basado en Casos, a continuación, se explican las características de los sistemas que se basan en este razonamiento.

1.3. Sistemas Basados en Casos

El Razonamiento Basado en Casos es un sistema experto que se basa en el proceso de solucionar nuevos problemas teniendo en cuenta las soluciones de problemas anteriores. Un abogado que apela a precedentes legales para defender alguna causa está usando Razonamiento Basado en Casos. También un ingeniero cuando copia elementos de la naturaleza, está tratando a esta como una “base de datos de soluciones”. El Razonamiento Basado en Casos es una manera de razonar haciendo analogías. Se ha argumentado que el Razonamiento Basado en Casos no sólo es un método poderoso para el razonamiento de computadoras, sino que es usado por las personas para solucionar problemas cotidianos. Más radicalmente, se ha sostenido que todo razonamiento es basado en casos porque está basado en la experiencia previa (10).

Ciclo del Razonamiento Basado en Casos

El proceso de dividir el Razonamiento Basado en Casos en diferentes subprocessos se puede particionar en cuatro procesos claramente diferenciados: (10).

1. **Proceso de recuperación** de los casos similares al que se analiza.
2. **Proceso de adaptación** de la información y el conocimiento que se tiene en este caso para resolver el problema.
3. **Proceso de evaluación** de la solución propuesta para el almacenamiento de la misma.
4. **Proceso de almacenamiento** de la parte de esta experiencia que pueda ser útil para la resolución de futuros problemas.

A continuación, se muestra la figura donde se representa el funcionamiento general de un sistema basado en casos.

Capítulo I: Fundamentación teórica de la investigación

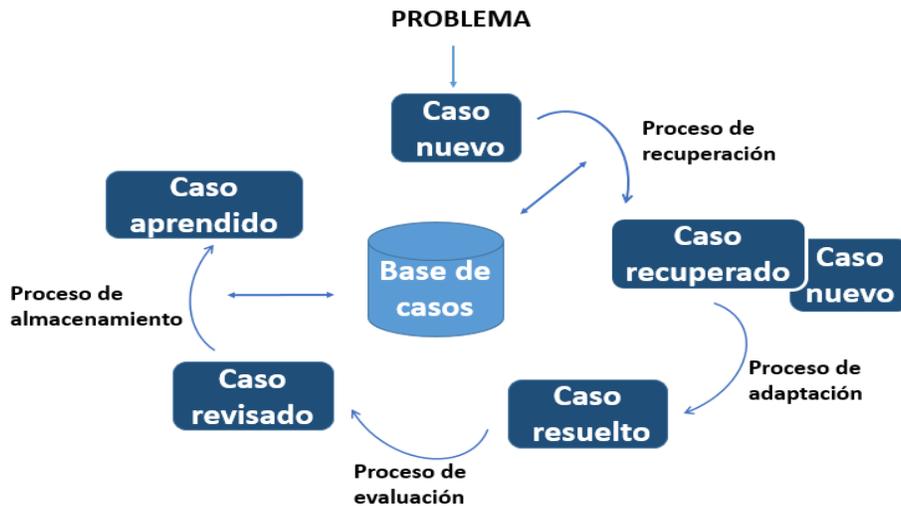


Figura 1: Ciclo básico de un sistema basado en casos (18).

Cuando se tiene un nuevo caso para revisar y dar una solución, el primer paso es, dado el caso base seleccionar los casos más semejantes, para esto se debe realizar una selección, a partir de los casos que se disponen en la base de casos. Una vez obtenido el conjunto de casos similares se debe adaptar la solución según el problema en cuestión. Finalmente retener la solución, almacenando en memoria el nuevo caso. Realizar un sistema basado en casos tiene grandes ventajas. A continuación, se presentan algunas de ellas.

1.3.1. Ventajas de un sistema basado en casos (11)

- Adquisición de conocimiento: la unidad básica del conocimiento es el caso. Los seres humanos por lo general articulan su conocimiento mediante ejemplos de problemas y soluciones anteriores (casos), más que por medio de reglas específicas y abstractas.
- Permite proponer soluciones a problemas rápidamente: esto lo logra ya que las respuestas no se derivan a partir de cero, sino de casos resueltos previamente.
- Aprendizaje: si la misma situación se presenta repetidamente, no se tiene que construir o generar la misma solución a partir de cero.
- Propone soluciones en dominios no entendidos completamente por el sistema.
- Ofrece un medio de evaluación de soluciones cuando no se cuenta con un método algorítmico.
- Se centra en las características o partes más importantes del problema.

A continuación, se realiza un análisis del sistema basado en casos SISDREA v1.0, donde se precisan aspectos que se pudieran mejorar.

Capítulo I: Fundamentación teórica de la investigación

1.4. Análisis de SISDREA v1.0.

SISDREA es un sistema basado en casos que realiza la revisión de recursos educativos para detectar posibles problemas de diseño. Este sistema obtiene los problemas de diseño, una vez introducido los metadatos de un caso base (recurso educativo digital que se quiere revisar), realizándose los procesos de recuperación y adaptación.

1.4.1. Proceso de recuperación en SISDREA v1.0

Este sistema comienza con el proceso de recuperación una vez el usuario introduce los datos referentes al caso base. Donde a través del algoritmo K-NN, utilizando la búsqueda por semejanza y de la distancia euclídea se realiza este proceso.

Identificados los rasgos de metadatos de un RED se procede al almacenamiento de estos, junto con la revisión del recurso, en una estructura de datos que favorezca su acceso cuando sea necesario. Dicha estructura será la encargada de representar los casos (Recursos Educativos Digitales) que van a conformar la base de casos.

Teniendo en cuenta que el sistema almacena casos sencillos, compuestos básicamente por metadatos de los recursos, se utiliza una representación plana de la base de conocimiento, la cual tiene varias ventajas, entre ellas se puede destacar que el almacenamiento de casos se realiza de forma muy sencilla. A continuación, se muestra la tabla 2 donde se presenta la lógica con la que esta estructurada la base de casos del sistema.

Tabla 2: Estructura de la base de casos del sistema (elaboración propia)

Caso base(X)	Rasgo predictor 1	Rasgo predictor 2	Rasgo predictor 3	...	Rasgo predictor n	
Caso (y)	Rasgo predictor y.1	Rasgo predictor y.2	Rasgo predictor y.3	...	Rasgo predictor y.n	Rasgo objetivo
Caso (z)	Rasgo predictor z.1	Rasgo predictor z.2	Rasgo predictor z.3	...	Rasgo predictor z.n	Rasgo objetivo
Caso(w)	Rasgo predictor w.1	Rasgo predictor w.2	Rasgo predictor w.3	...	Rasgo predictor w.n	Rasgo objetivo

Capítulo I: Fundamentación teórica de la investigación

Los rasgos predictores se obtienen a partir de clasificaciones definidas en estándar LOM, el cual es uno de los estándares más utilizados en la actualidad para la descripción de recursos educativos almacenados en distintos repositorios. LOM permite a través de una serie de metadatos caracterizar un objeto de aprendizaje garantizando su reutilización y obtención de forma fácil.

LOM utiliza las siguientes categorías para describir recursos educativos digitales (12):

- 1. General:** agrupa la información general que describe este recurso como un todo.
- 2. Ciclo de vida:** describe la historia y el estado actual de este recurso y aquellos que han afectado este recurso durante su evolución.
- 3. Metadatos:** describe la información específica sobre los propios metadatos (quién creó estos metadatos, cómo, cuándo, y con qué referencias).
- 4. Técnica:** describe los requerimientos técnicos y las características de este recurso.
- 5. Uso educativo:** describe las características educativas o pedagógicas claves de este recurso. Esta categoría almacena la información pedagógica esencial a aquellos involucrados en lograr una experiencia de aprendizaje de calidad. La audiencia incluye profesores, administradores, autores y estudiantes.
- 6. Derechos:** describe los derechos de propiedad intelectual y las condiciones de uso de este recurso.
- 7. Relación:** define las relaciones entre este recurso y otros recursos etiquetados. Relaciones múltiples pueden ser soportadas.
- 8. Anotación:** provee comentarios sobre el uso educativo de este recurso, quién creó esta anotación y cuándo.
- 9. Clasificación:** describe dónde se localiza este recurso dentro de un sistema de clasificación particular. Para definir múltiples clasificaciones, puede haber múltiples instancias de esta categoría.

Este proceso es el encargado de recuperar casos con características similares al caso analizado, siguiendo una estrategia para comparar parámetros de los rasgos predictores. Para ello, el sistema se conecta con el repositorio RHODA (9), a través del protocolo REST, del que obtiene los recursos educativos para realizar la comparación donde se emplean funciones de semejanza que determinan la similitud entre cada uno de los rasgos predictores de los casos comparados, las cuales se muestran a continuación (5):

Capítulo I: Fundamentación teórica de la investigación

Datos no numéricos: $sim(x, y) = \begin{cases} 1 & \text{si } x = y \\ 0 & \text{si } x \neq y \end{cases}$

Datos numéricos: $sim(x, y) = \frac{1-|x_i-y_i|}{\max(i)-\min(i)}$

Seguidamente, se efectúa una sumatoria de los valores de similitudes entre atributos y el resultado calculado es el equivalente a la distancia entre los casos analizados. Finalmente, se obtienen los cinco recursos educativos más semejantes.

1.4.2. Proceso de adaptación en SISDREA v1.0

Una vez obtenidos los cinco casos más semejantes comienza el proceso de adaptación, donde se realiza un análisis de los RED recuperados teniendo en cuenta el instrumento LORI, el cual se considera uno de los más adecuados para evaluar recursos educativos desde el punto de vista formativo y de diseño (5). Este instrumento brinda la posibilidad de evaluar recursos educativos a través de nueve criterios que indican los atributos a ser observados para la revisión de objetos de aprendizaje, además, define por cada categoría un conjunto de posibles problemas de diseño que pudiera presentar un recurso educativo. Permite al usuario dar una opinión según su apreciación del recurso, al expresar mediante una escala de cinco estrellas de valoración, su acuerdo o desacuerdo con los ítems propuestos para medir parámetros de calidad e idoneidad. LORI facilita la comparación entre RE al proporcionar un formato de revisión común (13).

Los indicadores de evaluación que propone este instrumento conllevan a su vez una serie de posibles problemas que pudieran atentar contra la integridad de cada uno de los propios indicadores, dichos problemas se encuentran definidos en el Manual de Usuario del Instrumento LORI (14) y son expuestos en el Anexo 1. SISDREA utilizará estos problemas, junto a la categoría más afectada, como salida del sistema (5).

Las categorías definidas por LORI son las siguientes:

1. Calidad de los contenidos
2. Adecuación de los objetos de aprendizaje
3. Adaptabilidad
4. Motivación
5. Diseño y presentación
6. Accesibilidad
7. Reusabilidad

Capítulo I: Fundamentación teórica de la investigación

8. Cumplimiento de estándares

9. Usabilidad

Por cada categoría se define un conjunto de clasificaciones, a las cuales, previamente aplicando preceptos de lógica difusa se obtuvieron valores de pertenencia a la clasificación mal. A continuación, se presentan cada una de estas clasificaciones con sus respectivos valores de pertenencia.

- Excelente: valor de pertenencia=0
- Muy bien: valor de pertenencia= 0.1
- Bien: valor de pertenencia = 0.3
- Regular: valor de pertenencia = 0.6
- Mal: valor de pertenencia = 1

Calculándose un promedio por cada una de las categorías de los cinco casos, la que tenga mayor promedio será la más afectada y teniendo en cuenta esto se define una serie de problemas de diseño que pudiera presentar el caso base.

1.4.3. Resultados del análisis de SISDREA v1.0

A partir del análisis realizado a SISDREA versión 1, se obtuvo los resultados siguientes.

- En el proceso de recuperación, al comparar los rasgos predictores, los atributos solamente toman valores 0 o 1, no permitiendo los valores en el rango comprendido entre ellos.
- No presenta una solución donde se le brinde al usuario información sobre los elementos que se tienen en cuenta para la detección de los problemas de diseño en recursos educativos, como son: la cantidad de recursos educativos que fueron analizados, los valores de los atributos del caso base y de los casos recuperados y el nivel de similitud que existe entre ellos; dificultándose la comprensión de los resultados brindados.
- La comunicación para obtener datos a utilizar en la recuperación de casos se puede establecer con un único repositorio de recursos educativos digitales a la vez y esta se debe establecer desde el código de la aplicación, por tanto, el usuario no cuenta con una interfaz donde se pueda realizar la gestión de esta comunicación de forma intuitiva.
- El sistema no permite guardar la solución para su posterior utilización, incumpliendo con uno de los procesos fundamentales de un sistema basado en casos, que es el proceso de almacenamiento.

Capítulo I: Fundamentación teórica de la investigación

- El sistema no está preparado técnicamente para permitir el intercambio de información (problemas de diseño de recursos educativos detectados) con otros sistemas que pudieran utilizarlos como entrada, lo que incide negativamente en capacidad de interoperabilidad del mismo.
- Dentro del ciclo general de un sistema basado en casos sólo se centra en los procesos de recuperación y adaptación.

Independientemente de las limitaciones que la versión 1 presenta, esta cuenta con elementos propios de su conceptualización e implementación que responden al propósito general para el que fue creada y que serán reutilizadas en la propuesta de solución. En la Tabla 3, se muestran cuáles de estos elementos sufrirán modificaciones en la nueva versión del sistema y cuáles se mantendrán, y a continuación, en el presente capítulo, se explicarán los que mayor significación aportan a la comprensión de la solución.

Tabla 3: Elementos que se mantienen y sufrirán cambios en base al sistema anterior (elaboración propia)

Proceso de recuperación		Proceso de adaptación	
Elementos que se mantienen	Elementos que sufrirán cambios	Elementos que se mantienen	Elementos que sufrirán cambios
LOM	Permitirá conectarse con varios repositorios a la vez.	LORI	-
Algoritmo K-NN	Fórmulas para comparar los rasgos predictores numéricos.	Lógica difusa	-
Búsqueda por semejanza			
Distancia euclídea.			
Protocolo REST para la conexión con repositorios.			

Capítulo I: Fundamentación teórica de la investigación

1.5. Algoritmos de recuperación de casos

Existen cantidades gigantescas de datos almacenados en bases de datos, almacén de datos y otros tipos de almacenes de información. Esta explosión de información ha provocado la aparición del KDD (*Knowledge Discovery from Databases*²) que es el proceso completo de extracción de conocimiento a partir de bases de datos. Dentro del KDD existe una etapa conocida como Minería de Datos, el objetivo de esta etapa es crear un modelo basado en los datos con el que se podrán hacer predicciones, entender mejor los datos o explicar situaciones pasadas. (15). Dentro del proceso de extracción de conocimiento existen algoritmos que se encargan de obtener la información necesaria de las bases de datos. A continuación, se hace referencia a uno de estos algoritmos.

Nearest Neighbor (K-NN)

El algoritmo K-NN clasifica cada dato nuevo en el grupo que corresponda, según tenga k vecinos más cerca de un grupo o de otro. Es decir, calcula la distancia del elemento nuevo a cada uno de los existentes, y ordena dichas distancias de menor a mayor para ir seleccionando el grupo al que pertenece. Este grupo será, por tanto, el de mayor frecuencia con menores distancias (16).

El K-NN es un algoritmo de aprendizaje supervisado, es decir, que a partir de un juego de datos inicial su objetivo es clasificar correctamente todas las instancias nuevas. El juego de datos típico de este tipo de algoritmos está formado por varios atributos descriptivos y un solo atributo objetivo (16).

En la situación descrita en la presente investigación, el algoritmo recibe como entrada los metadatos de un recurso educativo digital sin evaluación y realiza una búsqueda en la base de casos de recursos ya evaluados para encontrar los más semejantes. Posteriormente se toma la evaluación de estos para inferir la del recurso entrado (o caso base). Entiéndase por evaluación a la valoración del recurso, resultado del proceso de revisión del repositorio atendiendo a los indicadores del estándar LORI.

Tipos de búsqueda en un sistema basado en casos

Durante la recuperación de casos resulta necesario buscar qué casos son más semejantes al que se analiza, a continuación, se presenta la búsqueda por semejanza, la cual se utiliza en este sistema.

² Knowledge Discovery from Databases(KDD) proceso de extracción de conocimiento de bases de datos.

Capítulo I: Fundamentación teórica de la investigación

Búsqueda por semejanza

En la búsqueda por semejanzas, el objetivo es obtener el caso almacenado que más se parece al nuevo problema. Existen diversas formas de medir el grado de semejanza, que va desde expresiones analíticas de distancia hasta la descripción de algoritmos para obtener el grado de parecido entre dos objetos. Esto depende del tipo de información que se presente (17).

En este tipo de búsquedas las funciones de semejanza indican cómo se deben comparar los casos, pero es necesario saber qué se debe comparar. De todos los atributos que conforman un caso, es necesario definir cuáles son los más relevantes para efectuar la comparación (18).

1.5.1. Medidas de similitud

El algoritmo K-NN utiliza las distancias para la clasificación de un nuevo caso, seleccionando los casos que menos distancia tengan y por tanto serán los más semejantes. Para ello, se utiliza la distancia euclídea, a continuación se hará referencia al mismo.

Distancia euclídea

La distancia euclídea es la distancia "ordinaria" entre dos puntos de un espacio euclídeo, la cual se deduce a partir del teorema de Pitágoras. Esta distancia es la más común y usada en la implementación del método K-NN. Se define distancia Euclídea entre dos puntos X y Y como se muestra en la Fórmula, donde X_i es la coordenada de X en la dimensión i y Y_i es la coordenada de Y en la dimensión i (19).

$$d_E(x,y) = \sum_{i=1}^n \sqrt{x_i^2 + y_i^2}$$

Figura 2: Fórmula para determinar la distancia Euclídea (20).

Lógica difusa

La lógica difusa es una metodología que proporciona una manera simple y elegante de obtener una conclusión a partir de información de entrada vaga, ambigua, imprecisa, con ruido o incompleta. En general imita cómo una persona toma decisiones basada en información con las características mencionadas. Una de sus principales ventajas es la fácil integración con otras técnicas de la inteligencia artificial donde se maneje información con estas peculiaridades (21).

El adjetivo "difuso" se debe a que los valores de verdad utilizados tienen, por lo general, una connotación de incertidumbre. Lo difuso puede entenderse como la posibilidad de asignar más valores a los enunciados que los clásicos "falso" o "verdadero".

Capítulo I: Fundamentación teórica de la investigación

Además, de los elementos que componen los procesos de recuperación y adaptación se deben tener en cuenta otros que también son importantes para obtener una solución acertada.

En la actualidad una gran cantidad de recursos educativos digitales se encuentran a disposición de escolares y educadores en los centros educacionales. Influenciado por su impacto positivo, se ha originado una disposición hacia su apertura y democratización. Debido al aumento del uso de los mismos fue necesario disponer de diferentes repositorios que realicen la gestión de estos recursos. A continuación, se estará abordando sobre este tema.

1.6. Repositorios de recursos educativos digitales

La gestión de contenidos en los entornos *e-learning*³ están generando cambios en diversos sentidos. Entre estos cambios, los contenidos educativos se están construyendo como objetos de aprendizaje y se están recopilando en contenedores, conocidos como Repositorios de Objetos de Aprendizaje, que los organizan y los hacen accesibles para diferentes aplicaciones y perfiles de usuarios. Esto se logra fundamentalmente a través de la comunicación de aplicaciones interoperables y del intercambio de catálogos de metadatos compatibles. Como resultado de las propuestas de importantes grupos de estandarización, se ha dado paso a diferentes tecnologías y métodos de catalogación de recursos digitales, por lo que los proyectos de Repositorios de Objetos de Aprendizaje han ido en aumento (22).

Los repositorios de recursos educativos ofrecen una variedad de materiales didácticos en la red creados por entidades, docentes, investigadores y estudiantes. Estos repositorios pueden ser de objetos de aprendizaje, en cuyo caso se trata de unidades temáticas en red que desarrollan un contenido planteando el objetivo, el contenido y la evaluación (23).

Los recursos educativos residen en servidores específicos y utilizan determinados estándares y protocolos para su acceso. Son espacios especializados donde exclusivamente se pueden encontrar recursos educativos digitalizados, como pueden ser: recursos educativos digitales, objetos de aprendizaje, programas/software y otro tipo de aplicaciones. Pueden ser de distintos tipos dependiendo del área del conocimiento que traten (24).

En Cuba, la Red Universitaria de datos RedUniv y la Editorial Universitaria Cubana han desarrollado hace más de 10 años, diversas variantes de biblioteca virtual y repositorio de recursos educativos digitales para la Red de Universidades del Ministerio de Educación Superior. Actualmente, publica bajo

³ *e-learning* La plataforma de e-learning, es un espacio virtual de aprendizaje orientado a facilitar la experiencia de capacitación a distancia, tanto para empresas como para instituciones educativas.

Capítulo I: Fundamentación teórica de la investigación

el nombre de Biblioteca Virtual de la ECURED el repositorio de recursos educativos digitales del Ministerio de Educación Superior, disponible para todos. En este empeño participan las editoriales cubanas que han entregado parte de sus contenidos para su divulgación en la Web (25).

SISDREA v2.0 deberá conectarse con varios repositorios a la vez. Además, brindará una solución que permita la comunicación con otros sistema, para lograr esto se debe garantizar la interoperabilidad. A continuación, se hará referencia a la misma.

1.7. Interoperabilidad

La interoperabilidad es la habilidad de organizaciones y sistemas diversos para interactuar con objetivos consensuados, comunes y con la finalidad de obtener beneficios mutuos. La interacción implica que las organizaciones involucradas compartan información y conocimiento a través de sus procesos de negocio, mediante el intercambio de datos entre sus respectivos sistemas de tecnología de la información y las comunicaciones (26).

Para esta segunda versión de SISDREA se hace necesario asegurar la interoperabilidad con otros sistemas, ya que este debe permitir consumir servicios de los distintos repositorios de recursos educativos, además, una vez realizada la revisión de un RED, los problemas identificados deben estar accesibles para cualquier tipo de sistema. A continuación, se muestran las técnicas o tecnologías que hacen posible dicha interoperabilidad.

REST (Representational state transfer)

REST es un protocolo utilizado para la implementación de servicios web, el cual es una tecnología que utiliza precisamente un conjunto de protocolos y operaciones bien definidas que se aplican a todos los recursos de información.

Un servicio web es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre sistemas. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet (27).

REST presenta algunas características que hacen de él una buena opción para realizar la conexión con los repositorios de recursos educativos, algunas de dichas características son (27):

- Es muy ligero, sus respuestas contienen exactamente la información que se necesita.
- Son servicios fáciles de integrar con los sitios web existentes y están expuestos para que las páginas HTML los puedan consumir con la misma facilidad.

Capítulo I: Fundamentación teórica de la investigación

Además de utilizar el protocolo REST, este sistema tendrá que brindar la respuesta obtenida, luego de realizar la revisión de recursos educativos digitales. Con el objetivo que esta solución pueda ser utilizada por este sistema y en un futuro por cualquier otro, de forma fácil, se utilizarán archivos XML para el intercambio de la misma. A continuación, se hará referencia a este tipo de archivo.

XML (eXtensible Markup Language)

Los archivos XML, siglas en inglés de "*eXtensible Markup Language*" (lenguaje de marcas extensible), no hacen nada por cuenta propia. Simplemente son una forma de almacenar datos para que otros programas puedan leerlos fácilmente. Muchos programas usan XML para almacenar datos. Por lo tanto, se puede abrir, editar y crear un archivo XML con cualquier editor de texto (28).

Los expertos señalan varias ventajas que derivan de la utilización del XML, como por ejemplo que: es extensible (se pueden añadir nuevas etiquetas tras el diseño del documento); su analizador es estándar (no requiere de cambios para cada versión del metalenguaje); facilita el análisis y el procesamiento de los documentos XML creados por terceros (28).

Además de facilitar la organización de los recursos y la configuración de un programa, XML cumple un papel muy importante que es, sin lugar a dudas, su punto fuerte: le permite comunicarse con otras aplicaciones, de diferentes plataformas y sin que importe el origen de la información en común. Se pueden tener, por ejemplo, un programa corriendo en Windows con una base de datos de SQL Server, y otro en Linux con Oracle, ambos compartiendo datos gracias a una estructura en XML (29).

Como XML permite la compatibilidad entre sistemas para compartir información de una manera segura, fiable y fácil, este servirá como base luego de obtener los RED una vez consumido el servicio web REST publicado por los repositorios. Además, almacenará los problemas de diseño detectados, para que estos le sirvan a cualquier sistema que tenga que hacer uso de los mismos.

1.8. Lenguajes, herramientas y metodología a utilizar

Para la implementación de este sistema se hace necesario utilizar distintas metodologías, lenguajes y herramientas. A continuación, se presenta cada una de ellas y para qué se utilizan.

Lenguaje Java

Se utilizó Java en su versión 8.0 como lenguaje de programación orientado a objetos por las facilidades que ofrece, siendo actualmente uno de los lenguajes más usados para la programación en todo el mundo, además, fue el lenguaje utilizado en la primera versión de SISDREA del cual es necesario reutilizar los procesos de recuperación y adaptación. Ofrece una gran seguridad, escalabilidad y

Capítulo I: Fundamentación teórica de la investigación

portabilidad a las aplicaciones, además de ser multiplataforma y promover la reutilización de código (30).

Lenguaje JavaFX

Se utilizó JavaFX 3.0 para la creación de las interfaces del sistema ya que permite que sean más amigables y potentes que Java. Esta es una familia de productos y tecnologías de Oracle para la creación de aplicaciones web que tienen las características y capacidades de aplicaciones de escritorio, incluyendo aplicaciones multimedia interactivas. Las tecnologías incluidas bajo la denominación JavaFX son JavaFX Script y JavaFX Mobile. Las aplicaciones JavaFX pueden ser ejecutadas en una amplia variedad de dispositivos. En su versión (JavaFX 1.3, abril 2010) permite crear aplicaciones de escritorio, para celulares, la Web, TV, consolas de videojuegos, reproductores Blu-ray, entre otras plataformas, además puede integrarse código Java en programas JavaFX (31).

IntelliJ IDEA

Se utiliza IntelliJ IDEA 2017 para el desarrollo del software ya que permite el manejo de java incluyendo la integración con JavaFX para el visual con mayor facilidad. IntelliJ IDEA es un entorno de desarrollo integrado (IDE) para el desarrollo de programas informáticos. Es desarrollado por JetBrains (anteriormente conocido como IntelliJ), y está disponible en dos ediciones: comunidad y comercial (32).

Visual Paradigm

Esta herramienta se utiliza para el modelado del diseño de los componentes necesarios para el sistema. Este es un software de modelado UML que permite analizar, diseñar, codificar, probar y desplegar. Dibuja todo tipo de diagramas UML, genera código fuente a partir de dichos diagramas y también posibilita la elaboración de documentos.

Visual Paradigm for UML Enterprise Edition soporta: UML, SysML, ERD, BPMN, DFD, ArchiMate, diagramas, entre otros (33).

1.8.1. Metodología de Software

El desarrollo de software no es una tarea fácil; prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte, se tienen aquellas propuestas más tradicionales o robustas que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán, o sea, se basa en el uso exhaustivo de documentación durante todo el ciclo del proyecto. Por otra parte, se tienen las metodologías ágiles o

Capítulo I: Fundamentación teórica de la investigación

ligeras, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas (31). Seleccionar una metodología apropiada, resulta en ocasiones un factor esencial para obtener un software con la calidad esperada.

Metodología de software RUP

El Proceso Unificado Racional o RUP es un proceso de desarrollo de software desarrollado por la empresa *Rational Software*. Junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos (34).

RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Está incluido en el *Rational Method Composer* (RMC), que permite la personalización de acuerdo con las necesidades (34).

Metodología de software SCRUM

SCRUM es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos (35).

En SCRUM se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, SCRUM está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales (35).

Metodología de software XP

La Programación Extrema, es una metodología ligera de desarrollo de aplicaciones que se basa en la simplicidad, la comunicación y la realimentación del código desarrollado. Se considera el más destacado de los procesos ágiles de desarrollo de software. La programación extrema se diferencia de los métodos tradicionales, principalmente porque presenta más énfasis en la adaptabilidad que en la previsibilidad (36).

Capítulo I: Fundamentación teórica de la investigación

AUP-UCI (Proceso Unificado Ágil, versión UCI)

AUP: Es una versión simplificada de *Rational Unified Process* (RUP). Este define un flujo de trabajo con disciplinas diferentes a las de RUP, aunque conserva sus fases en cada una. La disciplina de modelación incluye la modelación del negocio, requisitos, análisis y diseño. Por otra parte, se integran además la Gestión de Cambios y Gestión de Configuración en una sola disciplina (37).

AUP-UCI se crea porque, no existía “una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, entre otros.) exigiéndose así que el proceso sea configurable”. Por lo que “se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI” (38).

- La adaptación de AUP que propone la actividad productiva de la UCI logra estandarizar el proceso de desarrollo de software, dando cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3.
- Redujo a uno la cantidad de metodologías que se usaban.
- De las cuatro fases que propone AUP (Inicio, Elaboración, Construcción, Transición), AUP-UCI decide mantener la fase de Inicio, pero modificando el objetivo de la misma, unifica las restantes tres fases de AUP en una sola, a la que llama Ejecución y agrega la fase de Cierre.

Además, en esta metodología se definen 4 escenarios (38):

- (Proyectos que modelen el negocio con CUN (Caso de Uso del Negocio) solo pueden modelar el sistema con CUS (Casos de Uso del Sistema).
- Proyectos que modelen el negocio con MC (Modelo Conceptual) solo pueden modelar el sistema con CUS (Casos de Uso del Sistema).
- Proyectos que modelen el negocio con DPN (Descripción de Proceso de Negocio) solo pueden modelar el sistema con DRP (Descripción de Requisitos por Proceso).
- Proyectos que no modelen negocio solo pueden modelar el sistema con HU (historias de usuario).

Capítulo I: Fundamentación teórica de la investigación

A continuación, se presenta una tabla comparativa de las metodologías ágiles: RUP, XP SCRUM y AUP-UCI.

Tabla 4: Comparación de metodologías (elaboración propia)

Criterios	RUP	XP	SCRUM	AUP-UCI
Tipo de framework	Análisis, diseño, implementación y documentación de sistemas orientados a objetos.	Basado en la adaptación, mayor flexibilidad, dinámica y funcional.	Gestión y desarrollo de software basados en un proceso iterativo e incremental.	Metodología adaptable a las condiciones del proyecto, análisis, diseño, implementación y documentación.
Tipo de revisión	En cada fase se realiza una o más iteraciones, perfeccionando así los objetivos. Si no se termina una fase no se continúa con la otra.	Se debe integrar como mínimo una vez al día y realizar las pruebas sobre la totalidad del proceso.	Sus pruebas se realizan al final del proceso enfatizando en la reutilización de los componentes de los programas comprobados.	Realiza varias iteraciones hasta que se cumplan perfectamente cada uno de los requisitos pertenecientes a dicha iteración.
Objetivos	Orientados a objetos que establecen las bases, plantillas y ejemplos para todos los aspectos y fases de desarrollo de software.	Basados en dar prioridad a trabajos con resultados directos, satisfacer al cliente, trabajar en grupos, actuar sobre variables.	Crea o redefine modelos existentes, alto rendimiento, ahorro de tiempo, reducción de los costos de desarrollo, mantiene la calidad en todo el desarrollo.	Aumentar la calidad del software que se produce. Las disciplinas propuestas por AUP: gestión de configuración, gestión de proyectos y entorno.

Capítulo I: Fundamentación teórica de la investigación

Tipo de desarrollo	Proceso iterativo incremental por fases (inicio, elaboración, construcción y transmisión)	Liviana, adaptable y desarrollado por fases (planificación del Proyecto, diseño, codificación y prueba)	Desarrollo simple que requiere trabajo duro. Control de forma empírica y adaptable.	Proceso iterativo e incremental por fases.
Facilidad de uso	Dirigido por casos de uso, establecimiento temprano de una buena arquitectura, iterativo e incremental, el trabajo se divide en mini proyectos.	Orientado para pequeños o medianos equipos, para proyectos con posibilidades de cambio, no apto para muchas personas.	No se basa en el seguimiento de un plan, modelo adaptable, construcción incremental basada en iteraciones, no existe trabajo con diseño o atracción.	Este escenario es muy conveniente si se desea representar una gran cantidad de niveles de detalles y la relaciones entre los procesos identificados.

Se decide escoger la metodología AUP-UCI ya que este sistema se desarrolla dentro de las líneas de investigación de GITAE⁴, donde se pretende que los artefactos que se generan respondan a las disposiciones establecidas para el desarrollo de proyectos en la UCI. Los sistemas que se han desarrollado para el propio entorno donde se implantará la solución utilizan esta metodología y, por ello, mantenerla garantizaría uniformidad en el proceso y en la documentación. Además, es una metodología híbrida, es decir, que no llega a ser robusta ni ágil, y al estar en un punto intermedio permite generar una documentación adecuada, ni muy extensa ni muy escasa, posibilitando hacer un sistema tan específico como deseen el usuario y el desarrollador. Esta metodología propone cuatro escenarios diferentes, garantizando que se aplique satisfactoriamente a cualquier tipo de proyecto, tanto aquellos que requieren un negocio bien definido como a los que no.

⁴ GITAE Grupo de Tecnologías para el Apoyo a la Educación

Capítulo I: Fundamentación teórica de la investigación

Debido a las características del proyecto y a que no se modela negocio, ya que está bien definido en el estudio realizado para la realización de SISDREA en su primera versión, solo se puede modelar la solución con historias de usuario, lo que representa el escenario cuatro. Además, el cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder desarrollarlos y probarlos. Es recomendado en proyectos no muy extensos, ya que una HU no debe poseer demasiada información.

1.9. Soluciones existentes a nivel internacional y nacional

En el ámbito internacional se han desarrollado un número importante de sistemas basado en casos, especializados en diferentes esferas como la cocina, los negocios, la medicina, el dominio de la ley, entre otros.

CHEF

CHEF es un planificador basado en casos que toma como entrada una conjunción de submetas que necesita lograr para conseguir un plan como salida. Su dominio es la creación de recetas. Las recetas son vistas como planes. Las recetas proporcionan la secuencia de pasos a seguir para preparar un plato. Por tanto, la entrada de CHEF son las metas que se pueden conseguir con las recetas (por ejemplo, incluye pescado, sofreír, sabor salado) y la salida es una receta (plan), que puede obtener esas metas. Como planificador basado en casos, CHEF crea sus planes a partir de viejos planes que funcionaron en situaciones similares y los modifica para adaptarlos a la nueva situación (39).

CASEY

CASEY es un sistema de diagnóstico basado en casos. Como entrada toma una descripción de un nuevo paciente, incluyendo muestras normales y presentándolas como síntomas. Su salida es una explicación causal de los desórdenes que tiene el paciente.

CASEY diagnostica los pacientes aplicando heurísticas de adaptación y emparejamiento basado en el modelo independientes del dominio y son tan precisas como el modelo del dominio en el que se aplican. La diagnosis la realiza en dos pasos: primero busca en la memoria casos y utiliza reglas de evidencia basadas en el modelo para determinar cuáles de los casos que ajustan parcialmente son suficientemente similares al nuevo problema para proporcionar una diagnosis precisa. Después aplica reglas de reparación basadas en el modelo (estrategias de adaptación) para adaptar el diagnóstico antiguo a la nueva situación (39).

Capítulo I: Fundamentación teórica de la investigación

HYPO

HYPO es un razonador interpretativo que trabaja en el dominio de la ley. HYPO toma como entrada una situación legal y como salida crea un argumento para sus clientes legales. HYPO puede crear argumentos igual de buenos tanto para la defensa como para la acusación. Típicamente en estos casos, un secreto de una compañía se desvela por otra compañía de la competencia y ésta saca provecho de la situación, por ejemplo, comercializando rápidamente un nuevo producto. Pero no todos los casos de revelación de secretos son ilegales y la tarea del programa es decidir, para cualquier caso, si es o no legal y crear un argumento que avale la decisión (39).

El estudio de estos sistemas permitió conocer que los mismos pueden ser utilizados en diferentes dominios de negocio, lo que evidencia la viabilidad de la utilización de este tipo de sistema en el aprovechamiento del conocimiento o la experiencia adquirida. Sin embargo, estos trabajos no presentan información concreta del funcionamiento interno del sistema, además, estas herramientas son propietarias y no existe suficiente información para obtener como fueron concebidos sus procesos. Para un análisis más profundo se tuvieron en cuenta sistemas desarrollados en el ámbito nacional, particularmente en la UCI que sí brindan información necesaria para este proceso.

En la UCI se han desarrollado sistemas expertos que se especializan en esferas como la medicina, la educación, el soporte y la gestión de procesos, entre otros. Para la presente investigación se consultaron algunos de ellos:

Sistema Basado en Casos para predecir la ocurrencia de reacciones adversas a medicamentos en la consulta médica (40).

Sistema basado en casos para contribuir a la disminución de los costos de la calidad de los proyectos del Centro de Telemática (41).

Sistema basado en casos para predecir la evaluación del estado de proyectos (42).

Sistema basado en casos para el trabajo didáctico con estudiantes de bajo rendimiento académico en la asignatura Matemática Discreta I” (43).

Sistema basado en el conocimiento para asistir el diagnóstico y sugerir tratamientos en pacientes hemipléjicos con alteraciones motora (44).

Sistema inteligente para la resolución práctica de problemas de Inteligencia Artificial utilizando la programación lógica (45).

Capítulo I: Fundamentación teórica de la investigación

Sistema basado en casos para la detección de problemas de diseño de recursos educativos digitales (5).

A continuación, se presenta una tabla comparativa de algunos sistemas basados en casos desarrollados en la Universidad de las Ciencias Informáticas (UCI). Teniendo en cuenta que se quiere estudiar el tipo de búsqueda y los procesos de recuperación y adaptación.

Tabla 5: Cuadro comparativo de sistemas basados en casos (elaboración propia)

Sistemas	Tipo de búsqueda	Recuperación	Adaptación
Sistema basado en casos para predecir la ocurrencia de reacciones adversas a medicamentos en la consulta médica.	Búsqueda por analogía	Árbol de decisión.	Para este sistema, la adaptación se efectúa determinando el caso de mayor utilidad esperada. Esto se realiza a través de la fórmula siguiente: $\mu(O_t) = \gamma\beta(O_0, O_t) + (1 - \gamma) * v_1(O_t)$ Fórmula para determinar la certidumbre total (46).
Sistema Basado en Casos para contribuir a la disminución de los costos de la calidad de los proyectos del Centro de Telemática.	Búsqueda por semejanza	Vecino más cercano. Distancia euclídea.	Ajuste de parámetros
Sistema Basado en Casos para predecir la evaluación del estado de proyectos.	Búsqueda por semejanza	Vecino más cercano.	El proceso de adaptación para esta investigación tiene como base la utilización de las técnicas de redes neuronales artificiales para buscar en la base de casos los más semejantes al caso entrado.

Capítulo I: Fundamentación teórica de la investigación

<p>Sistema basado en casos para el trabajo didáctico con estudiantes de bajo rendimiento académico en la asignatura Matemática Discreta I”</p>	<p>Búsqueda por semejanza</p>	<p>Vecino más cercano.</p>	<p>Para cumplir la regla del negocio de que los profesores de la asignatura tengan la posibilidad de escoger entre los cinco remediales propuestos, resultantes de la ejecución de la fase de recuperación, a su consideración el más adecuado para el nuevo problema. También se le brinda la posibilidad de crear un nuevo remedial, si considera que ninguno de los propuestos es adecuado o seleccionar uno de los existentes en el sistema.</p>
<p>Sistema Basado en el Conocimiento para asistir el diagnóstico y sugerir tratamientos en pacientes hemipléjicos con alteraciones motora.</p>	<p>Búsqueda por semejanza</p>	<p>Vecino más cercano. Distancia euclídea.</p>	<p>La solución obtenida en el paso anterior es analizada y ajustada en esta etapa. Para realizar esta operación se emplea un sistema de reglas que permite revisar los tratamientos propuestos, para identificar elementos que pueden haberse obviado o rasgos incorrectos que se hayan incluido en la solución. Las reglas incluyen relaciones entre síntomas y tratamientos, dependencia o solapamiento de ejercicios, prohibiciones y contraindicaciones, entre otras.</p>
<p>Sistema inteligente para la resolución práctica de problemas de Inteligencia Artificial utilizando</p>	<p>Búsqueda por semejanza</p>	<p>Recupera los vecinos cuya similitud sea superior al umbral especificado por el usuario.</p>	<p>En caso de que la clase sea nominal se busca el valor moda en los casos recuperados y en caso de que la clase sea de tipo numérica se calcula un promedio de los valores que poseen los casos recuperados.</p>

Capítulo I: Fundamentación teórica de la investigación

la programación lógica.			
-------------------------	--	--	--

De los sistemas analizados anteriormente, dos de ellos incluyen un módulo de explicación.

En el caso del sistema titulado: Sistema basado en el conocimiento para asistir el diagnóstico y sugerir tratamientos en pacientes hemipléjicos con alteraciones motoras (44), se muestran las instancias recuperadas, el nivel de semejanza que tienen respecto al nuevo caso y los valores que toman cada uno de sus atributos.

En el caso del sistema titulado: Sistema inteligente para la resolución práctica de problemas de Inteligencia Artificial utilizando la programación lógica (45), se muestran los casos recuperados, el método realizado para llegar a la solución y la similitud que existe entre el caso base y los elementos obtenidos.

El estudio del módulo de explicación desarrollado en estos sistemas permitió llegar a un consenso sobre los elementos que se deben mostrar en el módulo de este tipo que tendrá SISDREA v2.0. A partir de la comparación realizada a los sistemas desarrollados en la UCI, se evidenció que la tendencia en estos sistemas, es utilizar la búsqueda por semejanza para la obtención de los casos similares, para la recuperación de casos el método del vecino más cercano y para la adaptación, aunque no hay un método similar, en el caso de SISDREA v2.0 se utilizan técnicas de lógica difusa y el instrumento LORI.

Conclusiones parciales del capítulo

- A partir del análisis de SISDREA v1.0, se decidió mantener la utilización del algoritmo K-NN, la distancia euclídea, el estándar LOM e Instrumento LORI para la revisión de recursos educativos ya que son los más adecuados desde el punto de vista formativo y de diseño.
- El análisis de las características del Razonamiento Basado en Casos constató la necesidad de adicionar en la segunda versión de SISDREA funcionalidades que responden a los procesos de evaluación y almacenamiento del caso base.
- Se identificaron los elementos que se tienen en cuenta para la detección de problemas de diseño sobre los que se debe informar al usuario, así como el uso del protocolo REST y archivos XML para garantizar la interoperabilidad de la solución.
- Se seleccionó como metodología de desarrollo AUP-UCI, en su escenario 4.

Capítulo II: Propuesta de solución para SISDREA v2.0

Introducción

En el presente capítulo se precisan los elementos que conforman la propuesta de solución de la investigación, con el objetivo de esclarecer y comprender los aspectos que se tienen en cuenta para la realización del sistema basado en casos. Se detallan los artefactos como: especificación de requisitos funcionales y no funcionales, diseño de casos de prueba, historias de usuario, arquitectura de software, patrones de diseño y diagrama de clases del diseño, generados en la disciplina de requisitos y disciplina de análisis y diseño de la metodología de desarrollo seleccionada.

2.1. Descripción del sistema

Siguiendo las pautas para desarrollar la segunda versión del sistema basado en casos SISDREA e identificando los componentes fundamentales para que el diseño sea flexible, a continuación, se especifica el alcance del producto a desarrollar, para incrementar la comprensión, obtener buenos resultados de precisión y garantizar la interoperabilidad del sistema.

SISDREA v2.0 está formado por tres módulos, a continuación, se detallan las principales funcionalidades que se realizan en cada uno de ellos.

El primero de estos es el agente inteligente, en el cual, los procesos de recuperación y adaptación se mantienen de la versión anterior, aunque se le incluyen cambios que radican, fundamentalmente, en las funciones para la comparación de los rasgos predictores. Además, se le adicionan los procesos del ciclo de un sistema basado en casos: evaluación y almacenamiento de la solución propuesta.

El segundo módulo corresponde a la gestión de conexión con repositorios, que permite al usuario registrar y manipular los datos referentes a dichos repositorios. Además, los problemas de diseño detectados se almacenan en un archivo XML lo cual, le permite a cualquier sistema utilizar esta información.

El último módulo es el de explicación, su función es mostrar al usuario información sobre los elementos que se tienen en cuenta para la detección de los problemas de diseño en recursos educativos como: la cantidad de recursos educativos analizados, los metadatos de los recursos educativos recuperados y el caso base, y la similitud que existe entre ellos.

A continuación, se muestran de forma resumida las principales funcionalidades de cada uno de estos módulos.

Capítulo II: Propuesta de solución para SISDREA v2.0

Módulo de agente inteligente	Módulo de conexión con repositorios	Módulo de explicación
Introducir caso base. Solicitar recursos educativos. Recuperar los casos semejantes. Realizar adaptación. Evaluar la solución. Almacenar la solución. Mostrar problemas de diseño.	Insertar repositorio. Modificar repositorio. Eliminar repositorio(s). Mostrar listado de repositorios. Exportar listado de repositorios. Importar listado de repositorios.	Mostrar los elementos que se tienen en cuenta para la obtención de la solución propuesta. Mostrar ayuda.

Figura 3: Módulos que componen la solución propuesta (elaboración propia)

2.1.1. Módulo de agente inteligente

Como se mencionó anteriormente, para realizar la revisión de un recurso educativo, haciendo uso de un sistema basado en casos se deben realizar los procesos de recuperación y adaptación. En el proceso de recuperación se hicieron cambios en la forma de comparar los rasgos predictores, a continuación, en la tabla 6, se presentan los metadatos definidos por LOM que se utilizan en este sistema y, posteriormente, se presentan los cambios realizados para determinar la similitud de los mismos.

Tabla 6: Metadatos definidos por LOM que se utilizan en SISDREA (elaboración propia)

Sub-categoría/Categoría	Especificación	Valores
Estructura/General	Estructura interna del material	atómico, colección, networked (en red), jerárquico, lineal, mixta, ramificada y compartimentada
Nivel de agregación/General	Define la granularidad del material.	1,2,3,4
Estado/Ciclo de vida	Estado de producción del material	borrador, final, revisado, no disponible
Tipo de interactividad/Educacional	Tipo de interacción soportado por el material	activo, Indefinido, expositivo o mixto
Tipo de recurso de aprendizaje/Educacional	Especifica el tipo de material	ejercicio, simulación, cuestionario, diagrama,

Capítulo II: Propuesta de solución para SISDREA v2.0

		figura, gráfico, diapositiva, tabla, texto narrativo, examen, experimento, declaración de problema, valoración, lectura o multimedia
Nivel de interactividad/Educacional	Especifica el nivel de interacción del material	muy bajo, bajo, medio, alto o muy alto
Densidad semántica/Educacional	Una medida subjetiva de la utilidad educativa del material en comparación con su tamaño y/o duración	muy bajo, bajo, medio, alto o muy alto
Rol del usuario final/Educacional	Determina el papel del usuario final del material	profesor, autor, estudiante o gestor
Dificultad/Educacional	Grado de dificultad del material	muy fácil, fácil, medio, difícil o muy difícil
Propósito/Clasificación	Define el propósito de la clasificación	disciplina, idea, prerequisite, objetivo educativo, accesibilidad, restricciones, nivel educativo, nivel de habilidad, nivel de seguridad o competencia
Contexto/educacional	Se refiere al nivel que está dirigido	educación primaria, educación secundaria, educación superior, primer ciclo universitario, segundo ciclo universitario, postgrado, primer ciclo de escuela técnica, segundo ciclo de escuela técnica, formación profesional,

Capítulo II: Propuesta de solución para SISDREA v2.0

		formación continua, formación vocacional
Tipo/relación		es parte de, fue parte, es versión de, fue versión, es formato de, fue formato, referencia, es referenciado por, es basado en, es la base para, es requerido por, requiere

Funciones de comparación

Para este tipo de sistemas la forma de comparar cada rasgo es diferente y significativo dependiendo del grado de similitud que tengan entre ellos. Luego de un estudio de los rasgos predictores, la revisión de la semejanza que existe entre ellos y la consulta a expertos, se obtuvieron como resultado los valores que se muestran a continuación.

Por cada categoría se asignó un porcentaje de similitud, donde 0% representa que para los dos valores que se están comparando no son semejantes en lo absoluto y 100% significa que son completamente semejantes. A partir de los resultados obtenidos se obtuvo una función de comparación para cada uno de los rasgos predictores. A continuación, se muestra un ejemplo de estos resultados, el resto se encuentra en el anexo 2.

Dificultad: muy fácil, fácil, medio, difícil, muy difícil.

Tabla 7: Nivel de similitud de la categoría dificultad (elaboración propia)

	Muy fácil	Fácil	Medio	Difícil	Muy difícil
Muy fácil	100	75	50	25	0
Fácil	-	100	75	50	25
Medio	-	-	100	75	50
Difícil	-	-		100	75
Muy difícil	-	-			100

Capítulo II: Propuesta de solución para SISDREA v2.0

A continuación, se presenta un ejemplo de las funciones definidas a partir de la tabla anterior para este caso se escogió la categoría dificultad.

$$\delta(X(O_i), X(O)) = \begin{cases} 1 & \text{si } X(O_i) = X(O) \\ 0.25 & \text{si } (X(O_i) = \text{Muy fácil} \varphi X(O) = \text{Difícil}) \\ & \text{o } (X(O_i) = \text{Fácil} \varphi X(O) = \text{Muy difícil}) \\ 0.50 & \text{si } (X(O_i) = \text{Muy fácil} \varphi X(O) = \text{Medio}) \\ & \text{o } (X(O_i) = \text{Fácil} \varphi X(O) = \text{Difícil}) \\ & \text{o } (X(O_i) = \text{Medio} \varphi X(O) = \text{Muy difícil}) \\ 0.75 & \text{si } (X(O_i) = \text{Muy fácil} \varphi X(O) = \text{Fácil}) \\ & \text{o } (X(O_i) = \text{Muy fácil} \varphi X(O) = \text{Medio}) \\ & \text{o } (X(O_i) = \text{Medio} \varphi X(O) = \text{Difícil}) \\ & \text{o } (X(O_i) = \text{Difícil} \varphi X(O) = \text{Muy difícil}) \end{cases}$$

Figura 4: Funciones de similitud para la categoría dificultad (elaboración propia)

Una vez establecidas estas reglas para la comparación de los valores que pueden asociarse a los rasgos predictores, se garantiza una solución para el primer problema identificado en la situación problemática que define: las funciones de semejanza que se utilizan para la comparación de los rasgos predictores nominales solo permiten evaluaciones de 0 y 1 respectivamente.

Proceso de almacenamiento

Para garantizar que en un futuro el sistema pueda brindar soluciones a problemas más complejos este debe ir aprendiendo. Esto se traduce en incorporar nuevos casos a su base de conocimientos para poder reutilizarlos más adelante. El sistema ofrece la posibilidad de evaluar un recurso educativo a un usuario experto donde deberá introducir los metadatos de un caso base y darle una evaluación según las categorías de LORI, para almacenarlo en la base de conocimiento que posee el sistema. Además, brinda la posibilidad de almacenar los metadatos y evaluaciones de un recurso educativo que haya sido revisado. Garantizando que se solucione la debilidad detectada en la versión anterior, que plantea: el sistema no permite guardar la solución para su posterior utilización, incumpliendo con uno de los procesos fundamentales de un sistema basado en casos.

2.1.2. Módulo de conexión con los repositorios

Para la gestión de los repositorios, el sistema permitirá importar un archivo XML donde se encuentran almacenados los datos (nombre y dirección URL) de los repositorios con los que el sistema puede establecer la conexión. Luego se mostrará en una tabla dichos datos donde el usuario podrá eliminar, insertar y modificar estos datos. También, permite seleccionar uno o varios repositorios para establecer la conexión y obtener los datos necesarios para efectuar la revisión de los RED. En caso de no contar

Capítulo II: Propuesta de solución para SISDREA v2.0

con repositorios que brinden servicios para la obtención de recursos educativos, se podrá utilizar la base de datos local con que cuenta el sistema para realizar dicha revisión. Por último, se podrán exportar los datos de los repositorios almacenándolos en un archivo XML.

Una vez desarrollado este módulo e incorporado a SISDREA v2.0, permitirá solucionar las mejoras planteadas en la situación problemática donde se expone que no existe la posibilidad de configurar la conexión con repositorios distintos al que se encuentra previamente definido, ni se puede conectar con varios repositorios a la vez. Se logrará tener mayor acceso a repositorios de recursos educativos, posibilitando la obtención de resultados más certeros.

2.1.3. Módulo de explicación

Es necesario incluir para esta segunda versión de SISDREA un módulo de explicación que le permita al usuario tener conocimiento sobre los elementos que se tuvieron en cuenta para la obtención de la respuesta final del sistema, una vez realizada la revisión del recurso educativo. Cuando el usuario hace uso de la funcionalidad “Revisar recurso”, y el sistema le brinda una respuesta mediante el módulo de explicación, este puede observar el caso base, los casos recuperados, los niveles de semejanza que tienen respecto al nuevo caso, los valores que toman cada uno de sus rasgos predictores y la cantidad de casos recuperados. Con estos datos el usuario puede arribar a conclusiones sobre el basamento del proceso realizado para la detección de problemas de diseño en recursos educativos digitales.

Con la inclusión de este módulo a la primera versión de SISDERA se soluciona la limitación relacionada con la necesidad de brindarle al usuario información sobre el razonamiento llevado a cabo para la detección de problemas de diseño en recursos educativos.

Luego de conocer los elementos que serán modificados e incluidos a SISDREA v1.0, a continuación, se presenta una tabla donde se resumen las principales similitudes y diferencias entre las dos versiones del sistema en cuestión.

2.1.4. Similitudes y diferencias entre SISDREA v1.0 y SISDREA v2.0

En la tabla 8, se presenta un resumen donde quedan plasmadas las principales diferencias y similitudes de las dos versiones del Sistema Basado en Casos para la detección de problemas de diseño en recursos educativos.

Capítulo II: Propuesta de solución para SISDREA v2.0

Tabla 8: Similitudes y diferencias entre SISDREA v1.0 y SISDREA v2.0 (elaboración propia)

Similitudes y diferencias entre SISDREA v1.0 y SISDREA v2.0		
	SISDREA v1.0	SISDREA v2.0
Proceso de recuperación	Búsqueda por semejanza Algoritmo K-NN Distancia euclídea	Búsqueda por semejanza Algoritmo K-NN Distancia euclídea Con cambios en las funciones de semejanza.
Proceso de adaptación	Lógica difusa	Lógica difusa
Proceso de evaluación de la solución		Este proceso es el encargado de realizar la evaluación de la solución. Donde se define un umbral y si la similitud con el caso más semejante es menor que dicho umbral se decide almacenar la solución.
Proceso de almacenamiento	-----	Almacena la solución que no sea totalmente igual a una existente en la base de casos, y da la posibilidad a un experto de evaluar RED y almacenarlos en la base de datos local.
Interoperabilidad	Se puede conectar con un único repositorio a la vez. La conexión se establece a nivel de código.	Se puede conectar con varios repositorios a la vez, además presenta una interfaz que permite realizar la gestión de estos repositorios y se garantiza la interoperabilidad con otros sistema ya que

Capítulo II: Propuesta de solución para SISDREA v2.0

		permite exportar en un archivo los problemas identificados.
Otras funcionalidades adicionales al SBC	-----	Se adiciona un módulo de explicación. Permite a un experto evaluar recursos educativos para almacenarlos en la base de datos local.

Como se puede observar en la tabla comparativa, a SISDREA se le adicionarán nuevas funcionalidades que son importantes para obtener un sistema más completo y entendible para los clientes finales. La correcta identificación y descripción de los requisitos constituyen actividades esenciales para alcanzar estas metas. A continuación, se definen los requisitos funcionales y no funcionales que se deben garantizar para lograr los resultados esperados.

2.2. Especificación de requisitos del software

La especificación de requisitos del software es un producto que genera la ingeniería de requisitos. Describe la función, el desempeño de un sistema y las restricciones que regirán su desarrollo (47). Siguiendo la disciplina de requisitos que establece la metodología seleccionada, a continuación, se detallan los requisitos funcionales y no funcionales obtenidos una vez aplicados los métodos de obtención de requisitos.

Requisitos funcionales

Los requisitos funcionales (RF) son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de cómo se debe comportar ante situaciones particulares. En algunos casos, los requisitos funcionales pueden declarar también explícitamente lo que el sistema no debe hacer (48). A continuación, se describen los requisitos funcionales del sistema.

Tabla 9: Descripción de los requisitos funcionales (elaboración propia)

No.	Requisitos	Descripción	Prioridad	Complejidad
RF#1	Insertar repositorio	Esta funcionalidad debe permitir al usuario adicionar un nuevo repositorio a la lista de repositorios del sistema.	Media	Media

Capítulo II: Propuesta de solución para SISDREA v2.0

RF#2	Eliminar repositorio	Esta funcionalidad debe permitir al usuario eliminar el/los repositorio(s) seleccionado(s) de la lista de repositorios del sistema.	Baja	Baja
RF#3	Modificar repositorio	Esta funcionalidad debe permitir al usuario seleccionar un repositorio para modificar sus datos (nombre, dirección URL) para luego actualizar los datos en la lista de repositorios.	Baja	Baja
RF#4	Mostrar listado de repositorios	Esta funcionalidad debe permitir mostrar los repositorios que se encuentran almacenados en el archivo XML que se importa al sistema.	Media	Alta
RF#5	Importar listado de repositorios	Esta funcionalidad debe permitir importar la lista de repositorios que se encuentra guardada en un archivo XML.	Alta	Alta
RF#6	Exportar listado de repositorios	Esta funcionalidad debe permitir exportar la lista de repositorios a un archivo XML.	Media	Alta
RF#7	Mostrar información que se tuvieron en cuenta para la detección de problemas de diseño en RED	Esta funcionalidad debe permitir mostrar el proceso realizado para obtener la solución dada luego de revisar un recurso educativo.	Alta	Alta
RF#8	Solicitar recursos educativos a	Esta funcionalidad debe permitir luego de seleccionar los repositorios que se van a utilizar para revisar el recurso educativo	Alta	Alta

Capítulo II: Propuesta de solución para SISDREA v2.0

	los repositorios seleccionados	en cuestión, conectarse con dichos repositorios y solicitar los metadatos de los recursos educativos existentes.		
RF#9	Recuperar los recursos educativos semejantes al caso base	Esta funcionalidad debe permitir obtener de los repositorios seleccionados o de la base de datos local los metadatos de los recursos educativos semejantes al caso base.	Alta	Media
RF#10	Almacenar solución	Esta funcionalidad permite una vez evaluado un recurso educativo almacenar la solución propuesta en el archivo XML que funciona como base de datos local del sistema.	Alta	Media
RF#11	Mostrar ayuda.	Esta funcionalidad muestra cómo utilizar la aplicación.	Baja	Baja
RF#12	Evaluar recurso educativo digital por experto	Esta funcionalidad permite evaluar un recurso educativo digital según las categorías definidas por el instrumento LORI.	Alta	Media

Requisitos no funcionales

Los requisitos no funcionales pueden ser: de rendimiento, de software, de hardware, de seguridad, de apariencia o interfaz externa, entre otras. Debe pensarse en ellos como las características que hacen al producto atractivo, usable, rápido y confiable (49). En el desarrollo del sistema basado en casos SISDREA v2.0, se definieron requisitos funcionales y no funcionales que a continuación se listan:

Usabilidad: el sistema podrá ser usado por aquellas personas que no poseen conocimientos avanzados en informática. Permite el acceso a cada funcionalidad de forma fácil e intuitiva.

Capítulo II: Propuesta de solución para SISDREA v2.0

RNF#1: el sistema incluirá una ayuda que permite establecer los conocimientos básicos para la correcta utilización del sistema.

RNF#2: se mostrarán mensajes de confirmación con el objetivo de mantener informado al usuario de las acciones que ha realizado a partir de su interacción con el sistema.

RNF#3: se utilizará el color azul para resaltar elementos que son relevantes para la comprensión del usuario.

Apariencia e interfaz externa: el diseño de la interfaz será formal, serio y proporcionará claridad y correcta organización de la información, permitiendo la interpretación correcta e inequívoca de esta.

RNF#4: se mostrarán mensajes con un ícono de error de color rojo en caso de que el usuario realice alguna operación incorrecta.

RNF#5: se mostrarán mensajes con un ícono que representa una acción satisfactoria de color verde en caso de que el usuario realice alguna operación de forma correcta.

RNF#6: el sistema tendrá botones que estarán visibles solo en caso de ser necesarios, por ejemplo, el botón “Guardar cambios” en el módulo gestionar repositorios.

RNF#7: los elementos que son comunes están ubicados en el centro en la parte inferior del sistema, garantizando que el sistema sea intuitivo para el usuario.

RNF#8: se utilizarán los colores azul y blanco para el diseño de las interfaces.

RNF#9: los botones cambiarán de tamaño cuando el usuario ponga el cursor sobre ellos.

Software: debe mencionarse el software del que se debe disponer, después de implementado el sistema.

RNF#10: se requiere instalación previa de la Máquina Virtual de Java en su versión 1.8 o superior.

Hardware: se deben enunciar los elementos de hardware que se necesitan para que el software cumpla sus funcionalidades.

RNF#11: para el correcto funcionamiento de la aplicación se necesita una PC que tenga como mínimo las siguientes propiedades.

- RAM: 1 GB
- CPU: Intel Celeron 1.60GHz

Capítulo II: Propuesta de solución para SISDREA v2.0

A continuación, se presentan las historias de usuario para un mejor entendimiento de cómo funcionará el sistema una vez implementado cada uno de los requisitos planteados.

2.3. Historias de usuario (HU)

Una HU describe una funcionalidad que será útil para el usuario o cliente, de un sistema de software, muestran la funcionalidad que será desarrollada, pero no cómo se desarrollará (50). A continuación, se muestra un ejemplo de HU, el resto se encuentran en el anexo 3.

Tabla 10: Historia de usuario del requisito eliminar repositorio (elaboración propia)

Historia de Usuario	
Número: 3	Nombre del requisito: Eliminar repositorio.
Programador: Adriana Valdés Avilés	Iteración Asignada: 1
Prioridad: Baja	Tiempo Estimado: 3 días
Riesgo en Desarrollo: que no existan datos de los repositorios almacenados para realizar la eliminación.	Tiempo Real: 2 días
Descripción: Para eliminar algún repositorio el usuario debe acceder a la vista gestionar repositorios presionando en el botón " Gestionar repositorios". Luego de importar los repositorios que se encuentran almacenados en el archivo XML, el sistema mostrará una lista con los datos de los repositorios. El usuario selecciona el/los repositorio(s) que desea eliminar y presiona el botón eliminar. Se eliminan los datos de dicho(s) repositorio(s).	
Observaciones: en caso de que no existan datos de los repositorios para eliminar el sistema mostrará un mensaje informando que no existen repositorios para eliminar.	
Prototipo de interfaz:	

Capítulo II: Propuesta de solución para SISDREA v2.0

Gestionar conexión con repositorios		
Nombre	Dirección (URL)	Selección
Biblioteca Escolar Digital	http://www.bibliotecaescolardigitales	<input type="checkbox"/>
Buscador Infantil	http://www.buscadorinfantil.com	<input type="checkbox"/>
Educar	http://www.educarnet	<input type="checkbox"/>
Educacontic	http://www.educacontic.es	<input type="checkbox"/>
Educaed	http://www.educaed.net	<input type="checkbox"/>

Nombre:

Dirección:

2.4. Arquitectura de software

Según Pressman, la arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos. Las aplicaciones deben construirse con el empleo de capas en las que se tomen en cuenta distintas preocupaciones; en particular, deben separarse los datos de la aplicación de los contenidos de esta, y estos, a su vez, deben separarse con toda claridad del aspecto y la sensación de la interfaz (47).

En este sistema se utiliza la arquitectura en tres niveles. Que divide la aplicación en tres capas lógicas distintas, cada una de ellas con un grupo de interfaces perfectamente definido. La primera capa se denomina capa de presentación y normalmente consiste en una interfaz gráfica de usuario de algún tipo, la capa intermedia, o capa de empresa, consiste en la aplicación o lógica de empresa, y la tercera capa, la capa de datos, contiene los datos necesarios para la aplicación (51).

Estos tres niveles se separan en:

Capa de presentación: que engloba todas las clases relacionadas a la interfaz de usuario mediante la cual el mismo interactúa con el sistema.

Capa de negocio: está compuesta por las clases encargadas de realizar el procesamiento del sistema basado en casos.

Capa acceso a datos: en esta capa están las clases encargadas de la conexión con los repositorios, las que se encargan de almacenar los datos de estos y de los recursos educativos.

A continuación, se presenta la figura 5 donde queda plasmada la arquitectura del sistema.

Capítulo II: Propuesta de solución para SISDREA v2.0



Figura 5: Arquitectura en 3 niveles (elaboración propia)

2.5. Patrones de diseño:

Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias (52).

2.5.1. Patrones GRASP

Los patrones GRASP (Patrones Generales de Software para Asignar Responsabilidades en sus siglas en inglés General Responsibility Assignment Software Patterns) describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones (53).

Experto: este patrón responde a la pregunta: ¿cuál es el principio más básico para añadir responsabilidades a una clase? Asigna responsabilidades al experto de la información, es decir, a la clase que tiene la información necesaria para llevar la tarea a cabo (54). Se evidencia el patrón en la solución propuesta en las clases EntrarCasoController, RepositorioController, evaluarController e inicioController.

Capítulo II: Propuesta de solución para SISDREA v2.0

```
public class inicioController implements Initializable {
    Stage stage = new Stage();
    @FXML
    private Button repositorio;
    public Stage getStage() {
        return stage;
    } public void setStage(Stage stage) {
        this.stage = stage;
        stage.setOnCloseRequest(event -> Platform.exit());
    } public void initialize(URL location, ResourceBundle resources) }
```

Figura 6: Utilización del patrón Experto en la aplicación (elaboración propia)

Creador: este patrón responde a la pregunta: ¿quién debe ser el responsable de la creación de una nueva instancia de una clase? Plantea que debe recaer en la clase que agrega, contiene, registra, utiliza o tiene los datos de inicialización de la nueva instancia (54). Este patrón se evidencia en la clase GeneralController de la aplicación.

```
public GeneralController() {}
public REAs conexion(ArrayList<RepositorioModel> list) throws Exception {
    RestConectionUser reas = new RestConectionUser();
    reas_list = reas.getReas(list);
    return reas_list;
}
public String adaptacion() {
    AdaptationByFuzzyLogic adap = new AdaptationByFuzzyLogic();
    EvaluationResponseDescription adapt = adap.AdaptationByFuzzyLogic(similares,
cantRecuperar);
    String problemsResponse = adap.getProblemsResponse(adapt);
    return problemsResponse;
}}
```

Figura 7: Utilización del patrón Creador en la aplicación (elaboración propia)

Bajo Acoplamiento: este patrón responde a la pregunta: ¿cómo soportar baja dependencia e incrementar la reutilización? Propone asignar responsabilidades de tal manera que el acoplamiento sea el menor posible (54).

Capítulo II: Propuesta de solución para SISDREA v2.0

Este patrón se evidencia en la clase LomEstandar_Rasgos la cual contiene una instancia de la clase LomEstandar_Rasgo con lo cual toma todas las responsabilidades de la misma evitando una sobrecarga de relaciones con otras clases.

Alta Cohesión: este patrón responde a la pregunta: ¿cómo lograr que la complejidad sea lo más manejable posible? Propone asignar responsabilidades procurando que la cohesión sea lo más alta posible (54).

Este patrón se puede observar en la clase LomEstandar_Rasgo ya que sus funcionalidades son moderadas, están en su área funcional y colabora con otras clases como, por ejemplo: LomEstandar_SubCategory_General, LomEstandar_SubCategory_classification, LomEstandar_SubCategory_Educational, entre otras.

Controlador: este patrón responde a la pregunta: ¿quién debe manejar eventos del sistema? Propone asignar responsabilidades para el manejo de mensajes de eventos del sistema a una clase que represente al conjunto del sistema o negocio, representa algo del mundo real que está activo, o representa un administrador artificioso para todos los eventos del sistema (54).

```
public class Controller {
    public Controller() {}
    Stage stage = new Stage();
    public void mostrar(ActionEvent event) {
        try {
            Parent stage=
FXMLLoader.load(this.getClass().getResource("/resources/entrarCasoBase.fxml"));
            this.stage.setTitle("Caso Base");
            this.stage.setScene(new Scene(stage, 900.0D, 500.0D));
            this.stage.show();
        } catch (Exception var2) {
            var2.printStackTrace();    } } }
```

Figura 8: Utilización del patrón Controlador en la aplicación (elaboración propia)

2.5.2. Patrones GoF

Los patrones de la "pandilla de los cuatro" (GoF), describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos.

Capítulo II: Propuesta de solución para SISDREA v2.0

Singleton: permite el manejo de objetos únicos y que sean accesibles a otros objetos. Además, permite el acceso controlado a una única instancia.

La clase GeneralController representa un ejemplo del uso de este patrón en la implementación del sistema basado en casos, la cual es la clase controladora general, por lo que se decidió crear un único punto de acceso global hacia ella, implementando el patrón Singleton. Para ello se creó un constructor con visibilidad privada, y para ser instanciada se debe invocar a su método: getInstance, el cual es estático y retorna un atributo estático y privado que es del mismo tipo que la clase.

2.6. Diagrama de clases del diseño

Los diagramas de clases del diseño se encuentran conectados a la realización de un caso de uso o una historia de usuario. Son los encargados de mostrar las clases participantes, subsistemas y sus relaciones, así como, los atributos y operaciones correspondientes a cada clase (34).

A continuación, se muestran los diagramas de clases del diseño correspondiente a los módulos que se le realizaron cambios o se incluyeron a SISDREA v2.0, los cuales están resaltados de color azul y representan las clases y relaciones que contiene la aplicación.

Clases definidas para el proceso recuperación

La clase principal del módulo de recuperación es la llamada SimilarityEngine, dicha clase es el motor de inferencia del razonamiento del sistema basado en casos. Esta hace uso de varios objetos definidos como se muestra en la Figura 9. REAs es el objeto que contiene la colección de objetos REA (recurso) que constituyen la base de casos. Cada objeto REA posee una clase LoriEvaluation mediante la cual se gestionan las evaluaciones del recurso realizadas usando el instrumento LORI. Además, posee un objeto LomStandar_Rasgos, el cual constituye una colección de objetos LomStandar_Rasgo (rasgo de metadato) (5).

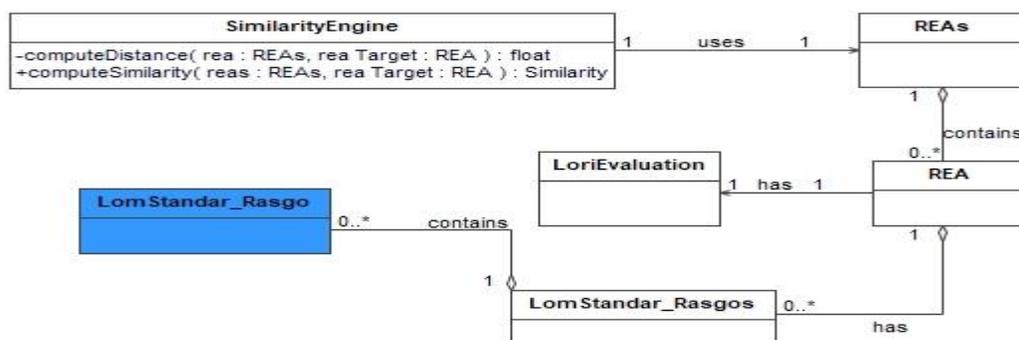


Figura 9: Clases definidas en el proceso de recuperación (5)

Capítulo II: Propuesta de solución para SISDREA v2.0

Para un mejor entendimiento de los cambios relacionados con el paquete LomStandar_Rasgo, a continuación, se presentan específicamente las clases que se modificaron dentro de este paquete, las cuales se encargan de realizar el pre-procesamiento de la información obtenida (dígase metadatos y evaluaciones de los recursos recuperados) y realizar la comparación de los mismos.

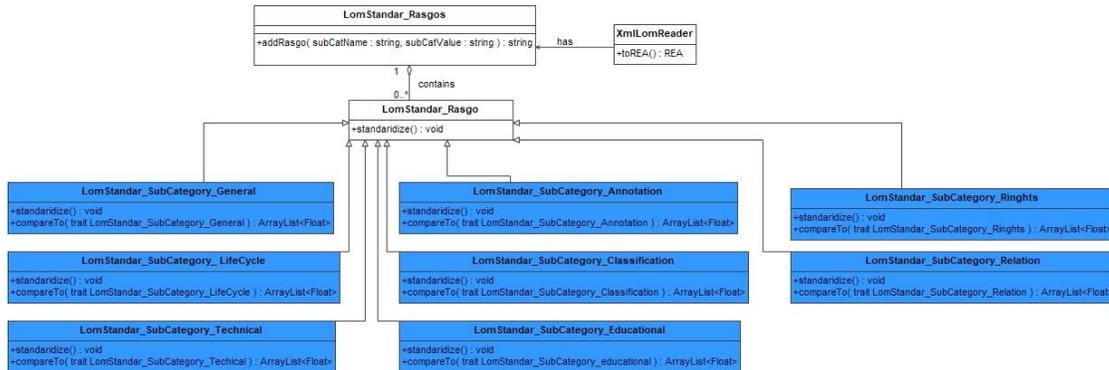


Figura 10: Relación entre las clases de la biblioteca LomStandarInterpreter (5)

Clases definidas en módulo conexión con los repositorios

Este módulo se encarga de realizar la gestión de los repositorios donde se podrá insertar, eliminar, modificar, importar y exportar datos referentes a los repositorios. Contiene una vista que realiza peticiones a la controladora (repositorio controller) esta a su vez se comunica con el modelo donde se almacenan los datos referentes a los repositorios para luego mandar la información a la vista.

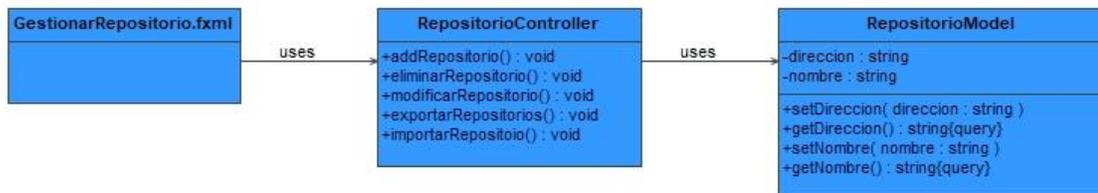


Figura 11: Diagrama de clases módulo de conexión con los repositorios (elaboración propia)

Por otro lado, se presenta el módulo encargado de realizar la conexión con los repositorios y dar la orden de revisar el recurso educativo digital.

Capítulo II: Propuesta de solución para SISDREA v2.0

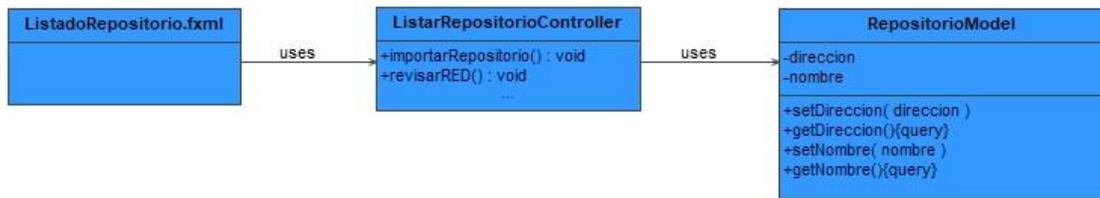


Figura 12: Diagrama de clases módulo listar repositorios (elaboración propia)

Clases definidas en el módulo explicación

A continuación, se presentan las clases que intervienen en el módulo de explicación.

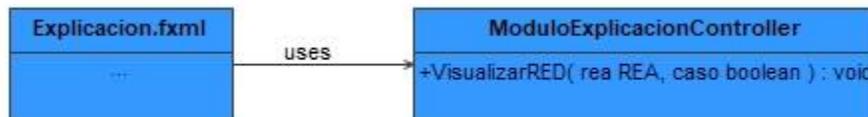


Figura 13: Diagrama de clases módulo de explicación (elaboración propia)

2.7. Diagrama de paquetes

Un paquete es un mecanismo utilizado para agrupar elementos de UML, es una parte de un modelo. Muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema. Cada paquete puede asignarse a un individuo o a un equipo, y las dependencias entre ellos pueden indicar el orden de desarrollo requerido (55).

Capítulo II: Propuesta de solución para SISDREA v2.0

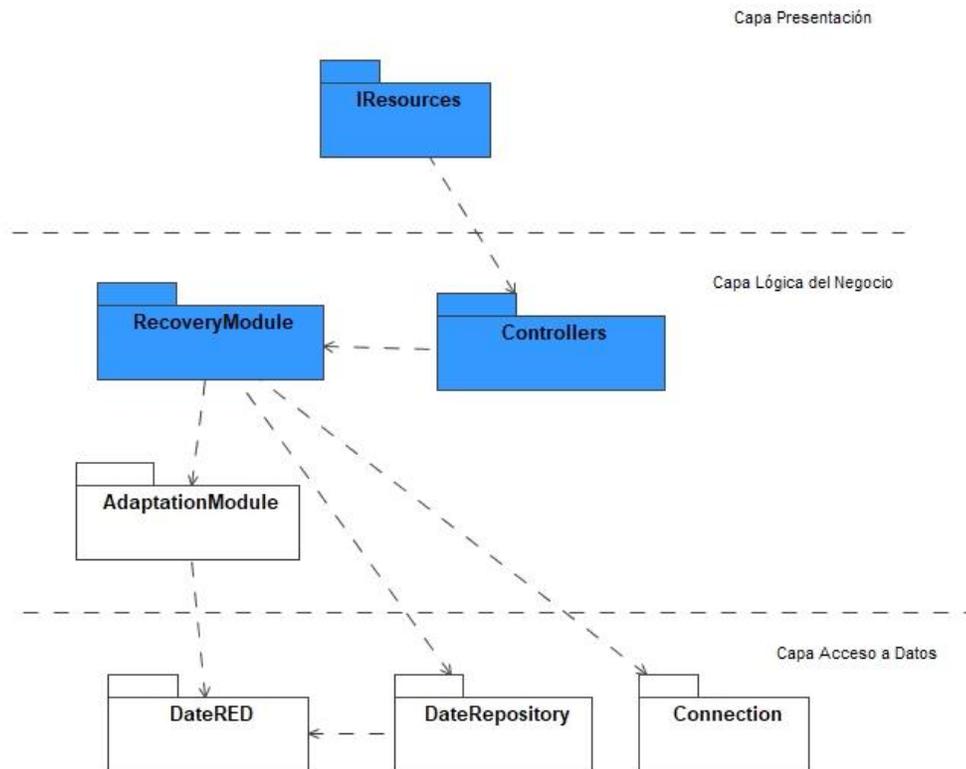


Figura 14: Diagrama de paquetes de SISDREA v2.0 (elaboración propia)

Conclusiones parciales del capítulo

- La descripción de la propuesta de solución, con la utilización de artefactos como la especificación de requisitos y las historias de usuario, garantizó un mejor entendimiento de los cambios propuestos para esta segunda versión de SISDREA.
- Los artefactos generados en el modelo de diseño sentaron las bases para la implementación y pruebas de los módulos.
- Se definió como arquitectura de software a emplear la Arquitectura en 3 niveles, logrando así una aplicación mucho más organizada y menos compleja para su entendimiento y futura evolución.

Capítulo III: Implementación y prueba de SISDREA v2.0

Introducción

En el presente capítulo se describen los elementos requeridos para desarrollar SISDREA v2.0 a partir de los resultados obtenidos en la disciplina análisis y diseño, dando inicio a la fase de implementación de la propuesta de solución. Se definen los estándares de codificación utilizados para el desarrollo del sistema basado en casos. También, se describen y realizan las pruebas de software aplicadas al sistema desarrollado, haciendo uso de los niveles de pruebas definidos por la metodología de desarrollo.

3.1. Estándar de codificación

Los estándares son lineamientos, directrices y protocolos que se establecen con el propósito de normalizar la escritura del código que conforma un desarrollo de software, procurando la consistencia y reusabilidad del mismo. Emplear un lenguaje claro al definir cada uno de los elementos de software a través de convenciones permite mantener el código de manera eficaz y eficiente, así mismo agiliza el proceso de detección de errores siempre que los estándares sean claros y se apliquen cabalmente (56).

Cada lenguaje de programación, tiene su propia convención sobre cómo se deben nombrar las variables, métodos y clases.

Estas convenciones o estándares de nomenclatura son un conjunto de normas para un lenguaje de programación específico y se recomiendan como buenas prácticas para facilitar la lectura del código y sea fácilmente entendible y mantenible (57).

Las más utilizadas son (57):

Camel Case: es un estilo de escritura que se aplica a frases o palabras compuestas. El nombre se debe a que las mayúsculas a lo largo de una palabra en CamelCase se asemejan a las jorobas de un camello. El nombre CamelCase se podría traducir como Mayúsculas/Minúsculas Camello. El término case se traduce como "caja tipográfica", que a su vez implica si una letra es mayúscula o minúscula y tiene su origen en la disposición de los tipos móviles en casilleros o cajas, y se puede dividir en dos tipos:

Upper Camel Case: se utiliza cuando la primera letra de cada una de las palabras es mayúscula. También denominado Pascal Case. Ejemplo: EjemploDeNomenclatura. Este estándar es utilizado

Capítulo III: Implementación y prueba de SISDREA v2.0

en la aplicación para nombrar las clases definidas en el sistema, por ejemplo LoriEvaluation, RepositorioControler, ModuloExplicacionController, entre otros.

```
public EntrarCasoController() {  
    LomStandar_Rasgos rasgos = new LomStandar_Rasgos();  
}
```

Figura 15: Utilización del estándar de codificación Upper Camel Case (elaboración propia)

Lower Camel Case: igual que la anterior con la excepción de que la primera letra es minúscula. Ejemplo: ejemploDeNomenclatura. Es muy usada en los lenguajes como Java, PHP, C#. Este patrón es utilizado para nombrar los métodos definidos en casa clase, por ejemplo: importarRepositorio, exportarRepositorio, modificarRepositorio, eliminarRepositorio, entre otros.

```
public void exportarRepositorio() throws IOException {
```

Figura 16: Utilización del estándar de codificación Lower Camel Case en la aplicación (elaboración propia)

3.2. Pruebas de software

Las pruebas de software consisten en la verificación dinámica del comportamiento de un programa, para asegurarse que arroja el resultado esperado y así garantizar la calidad del software. Además, representa una revisión final de las especificaciones de los requisitos del sistema. Este proceso tiene como objetivos (48):

- Demostrar al desarrollador y al cliente que el software satisface los requisitos.
- Descubrir defectos en el software en el que el comportamiento de este es incorrecto, no deseable o no cumple su especificación.

A continuación, se muestran los niveles de pruebas llevados a cabo una vez desarrollada la propuesta de solución.

- **Pruebas unitarias:** las pruebas unitarias se realizan sobre las funcionalidades internas de un módulo y se encargan de comprobar los caminos lógicos, ciclos y condiciones que debe cumplir el programa (46).
- **Pruebas del sistema:** las pruebas del sistema tienen como propósito principal ejercitar por completo el sistema basado en computadora, para verificar que los elementos del sistema se hayan integrado de manera adecuada y que se realicen las funciones asignadas. Para ello se aplican técnicas que se corresponden con el método de caja negra (54).

Capítulo III: Implementación y prueba de SISDREA v2.0

- **Pruebas de regresión:** las pruebas de regresión consisten en la repetición selectiva de pruebas para detectar fallos introducidos durante la modificación de un sistema o componente de un sistema. Se efectúan para comprobar que los cambios no han originado efectos adversos no intencionados o que se siguen cumpliendo los requisitos especificados (54).
- **Pruebas de aceptación:** el International Software Testing Qualifications Board⁵ (ISTQB) define la “Aceptación” como pruebas formales con respecto a las necesidades del usuario, requerimientos y procesos de negocio, realizadas para determinar si un sistema satisface los criterios de aceptación que permitan que el usuario, cliente u otra entidad autorizada pueda determinar si acepta o no el sistema (58).

3.2.1. Pruebas unitarias

Para evaluar las funciones internas del sistema, se utilizó el método de Caja Blanca, el cual utiliza una estructura de control descrita como parte del diseño al nivel de componentes para derivar los casos de prueba. Al emplear el método de Caja Blanca, el probador podrá derivar casos de prueba que (54):

- Garanticen que todas las rutas independientes dentro del módulo se han ejercitado por lo menos una vez.
- Se ejerciten los lados verdaderos y falsos de todas las decisiones lógicas.
- Se ejecuten todos los bucles dentro de sus límites operacionales.
- Se ejerciten estructuras de datos internos para asegurar su validez.

Para aplicar este método de prueba se empleó la técnica del Camino Básico, utilizando como ejemplo las funciones eliminarRepositorio e insertarRepositorio, pertenecientes a la clase RepositorioController, las cuales se desarrollan en las iteraciones 1 y 2, respectivamente.

La técnica del Camino Básico 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 pruebas derivados para ejecutar el conjunto básico deben garantizar que se ejecuta cada instrucción del programa por lo menos una vez durante la prueba (54). A continuación, se muestran los ejemplos de la numeración de los nodos definidos en cada porción del código.

⁵ ISTQB Comité Internacional de Certificaciones de Pruebas de Software.

Capítulo III: Implementación y prueba de SISDREA v2.0

Tabla 12: Numeración de los nodos en cada porción del código, método insertar (elaboración propia)

	<code>public void addRepositorio() {</code>
1	<code>RepositorioModel repositorio = new RepositorioModel; String texto = TFnombre.getText(); String textoD = TFdireccion.getText(); texto=texto.replaceAll("",""); textoD = textoD.replaceAll(" ", " ");</code>
2	<code>if (texto.length() == 0 textoD.length() == 0) {</code>
3	<code> MessageBox.show(stage.getOwner(), "Debe completar los campos" + " nombre y direccion", "error", MessageBox.OK MessageBox.ICON_ERROR); }</code>
4	<code>else { try {</code>
5	<code> new URL(TFdireccion.getText()).toURI();</code>
6	<code> repositorio.nombre.set(TFnombre.getText()); repositorio.direccion.set(TFdireccion.getText());</code>
7	<code> repositorios.add(repositorio);</code>
8	<code> TFnombre.setText(""); TFdireccion.setText(""); }</code>
9	<code> catch (Exception e) {</code>
10	<code> MessageBox.show(stage.getOwner(), "URL no válida", "error", MessageBox.OK MessageBox.ICON_ERROR); }</code>
11	<code> }}</code>

A continuación, se detallan los pasos realizados para aplicar la técnica Camino Básico en los métodos eliminar y adicionar datos de los repositorios, respectivamente:

Capítulo III: Implementación y prueba de SISDREA v2.0

1. **Confeccionar el grafo de flujo:** Utilizando el código representado en las tablas 11 y 12 se realizó la representación del grafo de flujo, el cual describe un flujo de control lógico y está compuesto por los siguientes elementos (54):

- **Nodos de gráfica de flujo:** Son círculos que representan una o más instrucciones procedimentales.
- **Aristas o enlaces:** Son flechas que representan el flujo de control y son análogas a las flechas del diagrama de flujo.
- **Regiones:** Son las áreas delimitadas por aristas y nodos.

En la Figura 17 y 18 se presentan los respectivos grafos de flujo obtenidos:

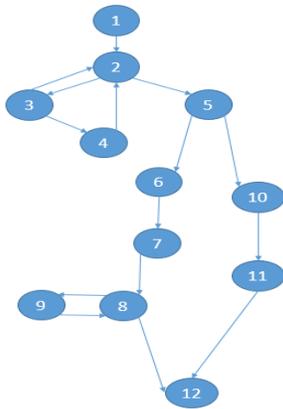


Figura 17: Grafo de flujo función eliminar

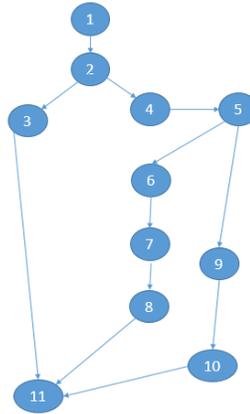


Figura 18: Grafo de flujo, función insertar repositorio

2. **Calcular la complejidad ciclomática:** Proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado define el número de caminos independientes del conjunto básico de un programa, y proporciona un límite superior para el número de pruebas que deben aplicarse para asegurar que todas las instrucciones se hayan ejecutado al menos una vez. La complejidad ciclomática se calcula mediante las tres formas siguientes (54).

$V(G)$ = cantidad de regiones

$V(G) = 4 - V(G) = A - N + 2$, donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.

$V(G) = P + 1$, donde P es el número de nodos predicados contenidos en el gráfico de flujo, es decir, nodos de donde emergen 2 o más aristas.

Capítulo III: Implementación y prueba de SISDREA v2.0

Tabla 13: Fórmulas y resultados por métodos de la complejidad ciclomática (elaboración propia)

Cantidad de regiones	$V(G) = A - N + 2$	$V(G) = P + 1$
Método eliminar	$(15 - 12) + 2 = 5$	$V(G) = 4 + 1 = 5$
Método insertar	$(12 - 11) + 2 = 3$	$V(G) = 2 + 1 = 3$

3. **Determinar un conjunto básico de rutas linealmente independientes:** El valor de $V(G)$ indica el número de rutas linealmente independientes de la estructura de control del programa, por lo que se definen las rutas independientes obtenidas:

Tabla 14: Tabla de rutas básicas (elaboración propia)

Caminos básicas	
Método eliminar	Método insertar recurso
Camino básico 1: 1-2-3-4-2-5-6-7-8-9-8-12	Camino básico 1: 1-2-3-11
Camino básico 2: 1-2-3-2-5-10-11-12	Camino básico 2: 1-2-4-5-9-10-11
Camino básico 3: 1-2-5-10-11-12	Camino básico 3: 1-2-4-5-6-7-8-11
Camino básico 4: 1-2-5-6-7-8-9-8-12	
Camino básico 5: 1-2-5-6-7-8-12	

4. **Obtención de casos de pruebas:** Cada camino independiente es un caso de prueba a realizar, de forma que los datos introducidos provoquen que se visiten las sentencias vinculadas a cada nodo del camino (54). En estos casos se obtuvieron cinco y tres caminos independientes en los métodos eliminar e insertar respectivamente, que dan lugar a la confección de igual número de casos de pruebas. A continuación, se muestran los casos de pruebas para cada uno de los métodos en cuestión:

Tabla 15: Caso de prueba eliminar: Ruta independiente #1 (elaboración propia)

Caso de prueba: eliminar correctamente un repositorio
Descripción: esta función debe eliminar de la tabla los datos de los repositorios que hayan sido seleccionados previamente, mostrando luego los restantes repositorios que quedan en el sistema.

Capítulo III: Implementación y prueba de SISDREA v2.0

Entrada	Tabla con los datos de los repositorios con que cuenta el sistema.
Resultados esperados	Se mostrará una tabla con los datos de los repositorios restantes luego de realizada la eliminación.
Condiciones	Debe existir al menos un repositorio para eliminar y debe estar seleccionado.

Tabla 16: Caso de prueba eliminar: Ruta independiente #2 (elaboración propia)

Caso de prueba: alerta de selección de repositorios para eliminar.	
Descripción: esta función debe eliminar de la tabla de repositorios los que hayan sido seleccionados previamente, mostrando luego los restantes repositorios que quedan en el sistema.	
Entrada	Se muestran los datos de los repositorios pero no se ha seleccionado ninguno para eliminar.
Resultados esperados	Debe mostrar un mensaje donde le informe al usuario que no se ha seleccionado ningún repositorio para realizar la eliminación.
Condiciones	Deben existir repositorios almacenados, sin embargo no se realiza la selección de ninguno de ellos.

Tabla 17: Caso de prueba eliminar: Ruta independiente #3 (elaboración propia)

Caso de prueba: no existen repositorios en el Sistema.	
Descripción: esta función debe eliminar de la tabla de repositorios los que hayan sido seleccionados previamente, mostrando luego los restantes repositorios que quedan en el sistema.	
Entrada	No existen datos de los repositorios en el sistema.
Resultados esperados	Debe mostrar un mensaje donde le informe al usuario que no se ha seleccionado ningún repositorio para realizar la eliminación.

Capítulo III: Implementación y prueba de SISDREA v2.0

Condiciones	No deben existir repositorios almacenados.
-------------	--

Tabla 18: Caso de prueba eliminar: Ruta independiente #4 y #5 (elaboración propia)

Caso de prueba: camino incorrecto	
Descripción: esta función debe eliminar de la tabla de repositorios los que hayan sido seleccionados previamente, mostrando luego los restantes repositorios que quedan en el sistema.	
Entrada	No existen datos de los repositorios.
Resultados esperados	El sistema no debe seguir estas rutas puesto que, al no tener repositorios almacenados no se podrá continuar con la eliminación en ninguno de los casos.
Condiciones	No deben existir repositorios almacenados.

Tabla 19: Caso de prueba insertar repositorio: Ruta independiente #1 (elaboración propia)

Caso de prueba: completar campos.	
Descripción: esta función debe insertar un nuevo repositorio	
Entrada	No se completan los campos nombre y dirección.
Resultados esperados	El sistema debe mostrar un mensaje informando que se deben completar los datos nombre y dirección.
Condiciones	No se deben completar los datos referentes al nombre y la dirección del repositorio.

Tabla 20: Caso de prueba insertar repositorio: Ruta independiente #2 (elaboración propia)

Caso de prueba: URL no válida.	
Descripción: esta función debe insertar un nuevo repositorio	
Entrada	Se introducen los datos de un nuevo repositorio pero la dirección URL no es correcta.

Capítulo III: Implementación y prueba de SISDREA v2.0

Resultados esperados	El sistema debe mostrar un mensaje informando que la URL introducida no es correcta.
Condiciones	Se deben introducir los datos nombre y URL, pero dirección no está de la forma correcta.

Tabla 21: Caso de prueba insertar repositorio: Ruta independiente #3 (elaboración propia)

Caso de prueba: insertar repositorio.	
Descripción esta función debe insertar un nuevo repositorio.	
Entrada	Se completan correctamente los datos referentes al nombre y la dirección del repositorio.
Resultados esperados	El sistema debe insertar los datos de un nuevo repositorio de recursos educativos a la lista de repositorios que posee el sistema.
Condiciones	Se debe haber completado los datos nombre y URL correctamente.

Descripción de los casos de prueba para el método eliminar:

Caso de prueba eliminar repositorio de forma correcta: se probó que al seleccionar el/los repositorios que se quieren eliminar, se presiona el botón "Eliminar" y se sustraen dichos repositorios seleccionados, quedando solo los que no fueron seleccionados para ser eliminados, lo que evidencia que el caso de prueba se ejecutó favorablemente.

Caso de prueba alerta de selección de repositorios para eliminar: para proporcionar la ejecución de este camino, deben existir repositorios almacenados en el sistema, sin embargo, no se selecciona ninguno de ellos para su posterior eliminación, donde el sistema muestra un mensaje alertando que debe seleccionar algún repositorio para eliminar, siendo esta la respuesta esperada en este escenario.

Caso de prueba no existen repositorios en el sistema: para este caso no deben existir repositorios almacenados en el sistema y luego de presionar el botón eliminar se mostrará un mensaje informando que se debe seleccionar al menos un repositorio, siendo esta la respuesta esperada para este camino.

Capítulo III: Implementación y prueba de SISDREA v2.0

Caso de prueba camino incorrecto: para estos casos al no tener repositorios almacenados, no se podrá seguir con estos caminos, el sistema al comprobar que no tiene repositorios almacenados en la lista de seleccionados no entrará en la condición if y por tanto no se realizará la operación para ninguna de las dos rutas propuestas, demostrando así que el sistema brinda la respuesta esperada.

Descripción de la ejecución de las pruebas para el método insertar:

Caso de prueba completar campos: se probó que al no completar los datos referentes a un repositorio que se desea insertar, el sistema muestra un mensaje informando que “debe completar los campos nombre y dirección”, mostrando los resultados esperados, lo que evidencia que el caso de prueba se ejecutó favorablemente.

Casos de prueba URL no válida: en este caso se probó que si el usuario introduce el nombre y la dirección URL pero esta última no tiene la forma {http/https/ftp}://(...) el sistema muestra un mensaje informando “URL no válida”, Siendo esta la respuesta esperada en estos escenarios.

Caso de prueba insertar repositorio: una vez introducidos correctamente los datos nombre y dirección URL referentes al repositorio que se desea insertar, el sistema adiciona los mismos a la lista de repositorio, demostrando así que el sistema responde de la manera esperada.

Resultados: Una vez ejecutados todos los casos de pruebas obtenidos a través de la aplicación de la técnica ruta básica, se realizaron dos iteraciones donde se obtuvo en la primera iteración cuatro No Conformidades (NC), las cuales fueron solucionadas y en la segunda iteración no se encontraron NC. Por lo que se puede concluir que luego de realizar el método de Caja Blanca, donde se diseñaron y ejecutaron los casos de prueba correspondientes, se logró asegurar el cumplimiento del proceso de mejora del código.

Las no conformidades se centraron en errores funcionales.

NC#1: el sistema permite insertar un nuevo repositorio con una dirección URL incorrecta. (Alta)

NC#2: al presionar el botón eliminar no se sustrae correctamente el repositorio deseado. (Media)

NC#3: no permite eliminar varios repositorios a la vez. (Baja)

NC#4: al presionar el botón revisar sin introducir la cantidad de recursos educativos que se deben recuperar el sistema no muestra un mensaje indicando que se debe introducir una cantidad de RED. (Alta)

Capítulo III: Implementación y prueba de SISDREA v2.0

3.2.2. Pruebas de sistema

Como pruebas para evaluar las funcionalidades externas del componente, se utilizó el método de Caja Negra, el cual se centra en los requisitos funcionales del software. Este permite al probador derivar un conjunto de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales de un programa. (54).

Con el objetivo de aplicar el Método de Caja Negra, es necesario apoyarse en los Diseños de Casos de Prueba (DCP) propuestos por la metodología de desarrollo de software seleccionada. Un DCP es un artefacto generado en la disciplina Análisis y Diseño, utilizado en la disciplina de Pruebas Internas para llevar a cabo las pruebas funcionales. Tiene como objetivo comprobar el correcto funcionamiento del sistema, en estos se incluyen las entradas, resultados y condiciones con la que se ha de verificar, constituyendo la guía principal para el probador (48).

Para el diseño de estos casos de pruebas se tuvo en cuenta la técnica de Partición Equivalente. Esta técnica divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Es por esto que por cada requisito se obtuvo la descripción de las variables (DV), para luego dividir el campo de entrada en clases de datos válidos e inválidos. Una vez definida la DV, se procede a diseñar los escenarios de pruebas para cada caso de prueba. A continuación, se muestra un ejemplo de los casos de prueba aplicados al sistema, en este caso se muestra el caso de prueba de la funcionalidad insertarRepositorio.

Tabla 22: Caso de prueba insertar repositorio (elaboración propia)

Diseño de caso de prueba RF: insertar repositorio.						
Descripción general:						
Permite insertar los datos de un nuevo repositorio al XML que contiene el sistema para almacenar estos datos.						
Escenario	Descripción	Nombre	URL	Respuesta del sistema	Flujo central	
1.1 Insertar datos de un repositorio correctamente	El usuario introduce los datos nombre y dirección	V	V	2. El sistema muestra la sesión "Gestionar repositorio"	1. El usuario presiona el botón "gestionar repositorios" ubicado en la vista principal del sistema.	

Capítulo III: Implementación y prueba de SISDREA v2.0

	URL del repositorio que se desea incluir.			5. El sistema incluye correctamente los datos de un nuevo repositorio.	3. El usuario introduce los datos del repositorio que desea incluir de forma correcta. 4. El usuario presiona el botón "Insertar".
1.2 Insertar repositorio con dirección no válida	El usuario introduce los datos nombre y una dirección URL incorrecta.	V	I	2. El sistema muestra la sesión "Gestionar repositorio" 5. El sistema muestra el mensaje "dirección URL incorrecta".	1. El usuario presiona el botón "gestionar repositorios" ubicado en la vista principal del sistema. 3. El usuario introduce el nombre y una URL de forma incorrecta. 4. El usuario presiona el botón "Insertar".
1.3 Insertar repositorio con el campo nombre vacío.	El usuario introduce la dirección URL pero deja en blanco el campo nombre del repositorio.	I	V	2. El sistema muestra la sesión "Gestionar repositorio" 5. El sistema muestra el mensaje de error " el campo nombre no puede estar vacío".	1. El usuario presiona el botón "gestionar repositorios" ubicado en la vista principal del sistema. 3. El usuario introduce la dirección URL pero deja en blanco el campo nombre. 4. El usuario presiona el botón "Insertar".

Capítulo III: Implementación y prueba de SISDREA v2.0

<p>1.4 Insertar repositorio con el campo dirección URL vacío.</p>	<p>El usuario introduce el nombre del repositorio pero deja en blanco el campo dirección URL</p>	<p>V</p>	<p>I</p>	<p>2. El sistema muestra la sesión “Gestionar repositorio”</p> <p>5. El sistema muestra el mensaje de error “ el campo dirección URL no puede estar vacío”.</p>	<p>1. El usuario presiona el botón “gestionar repositorios” ubicado en la vista principal del sistema.</p> <p>3. El usuario introduce el nombre del repositorio pero deja en blanco el campo dirección URL.</p> <p>4. El usuario presiona el botón “Insertar”.</p>
<p>1.5 Insertar repositorio sin introducir ningún dato.</p>	<p>El usuario no introduce los datos nombre y dirección URL y presiona el botón “Insertar”</p>	<p>I</p>	<p>I</p>	<p>2. El sistema muestra la sesión “Gestionar repositorio”</p> <p>5. El sistema muestra el mensaje de error “debe completar los campos nombre y dirección”.</p>	<p>1. El usuario presiona el botón “gestionar repositorios” ubicado en la vista principal del sistema.</p> <p>2. El usuario presiona el botón “Insertar”.</p>

Después de realizadas las pruebas al sistema, fueron detectadas las siguientes No Conformidades (NC) que se clasifican según el defecto detectado teniendo en cuenta el tipo de no conformidad y el impacto de la misma.

Capítulo III: Implementación y prueba de SISDREA v2.0

- **Funcionalidad**

NC#1: no se mostraban todos repositorios que se encontraban almacenados en el sistema. (alta)

NC#2: no se conocía el total de recursos educativos almacenados en la base de datos local. (media)

NC#3: no permitía revisar un recurso educativo sin utilizar un repositorio. (alta)

- **Opción que no funciona:**

NC#4: la opción seleccionar RE de la base de datos local no funcionaba. (alta)

- **Ortografía:**

NC#5: omisiones de tildes y cambio de mayúscula por minúscula. (baja)

Resultados: De un total de 12 casos de prueba, tres de ellos resultaron no satisfactorios, lo cual representa el 21.42% del total de casos de pruebas realizadas. Mientras que los 10 restantes resultaron satisfactorios para un 78.57%, como se muestra en la figura. Los errores detectados por los casos de pruebas no satisfactorios fueron solucionados.



Figura 19: Resultados obtenidos en las pruebas de caja negra (elaboración propia)

3.2.3. Pruebas de regresión

Con el objetivo de verificar que se solucionaron las no conformidades detectadas y que la solución de estas, no provocaron nuevos errores, durante el desarrollo de las pruebas unitarias y funcionales, se realizaron las pruebas de regresión. En este caso el resultado fue el esperado, es decir no se detectaron NC.

Capítulo III: Implementación y prueba de SISDREA v2.0

3.2.4. Prueba de aceptación

Las pruebas de aceptación también llamadas pruebas del cliente, son especificadas por el cliente y se enfocan en las características y funcionalidades generales del sistema que son visibles y revisables por parte del cliente. Estas pruebas se realizan a partir de las historias de los usuarios que se han efectuado como parte de la liberación del software (54).

Al finalizar cada una de las iteraciones se desarrollaron pruebas de aceptación donde se identificaron en una primera iteración cuatro no conformidades de estas una de complejidad alta dos de complejidad media y una de complejidad baja. En la segunda iteración el número de no conformidades se redujo notablemente. Una vez resueltas las no conformidades el cliente firma el acta de aceptación del sistema, la cual se encuentra en el Anexo 4.

NC#1: el botón insertar no insertaba correctamente. (alta)

NC#2: no se mostraba el nombre del recurso educativo revisado por el sistema, en el vista encargada de mostrar los elementos que se tuvieron en cuenta para llegar a la solución propuesta. (media)

NC#3: el campo de texto donde el usuario introduce la cantidad de recursos educativos a recuperar permitía incluir letras. (media)

NC#4: existían errores ortográficos por ejemplo la falta de tildes y mayúsculas. (baja)

Conclusiones parciales del capítulo

- La definición de los estándares de codificación permitió normalizar la escritura del código que conforma el desarrollo del sistema, garantizando la consistencia y reusabilidad del mismo.
- Las pruebas realizadas permitieron detectar y corregir un grupo de no conformidades y obtener la aceptación del cliente; comprobándose que el sistema responde satisfactoriamente a las necesidades identificadas.

Conclusiones

Con el desarrollo de la presente investigación se arriba a las siguientes conclusiones:

El análisis de la primera versión de SISDREA permitió conocer las principales deficiencias que presenta este sistema, así como los elementos que se deben incluir para obtener un sistema más usable por los usuarios finales.

El estudio de los sistemas similares permitió obtener los principales elementos a incluir en el módulo de explicación del sistema.

Las herramientas, tecnologías y lenguajes descritos en el capítulo 1, soportaron el desarrollo de la propuesta de solución, que estuvo guiado por los artefactos generados en correspondencia con la Metodología AUP-UCI.

Se obtuvo, como producto final, una segunda versión de SISDREA, que garantiza una correcta comprensión del usuario, y permite conseguir una mayor precisión en los resultados obtenidos en la detección de problemas de diseño en recursos educativos, dando cumplimiento con el objetivo general trazado.

Con la inclusión del módulo de conexión a los repositorios se garantiza la gestión de los repositorios de forma fácil y, además, se obtiene una base de datos más amplia, mostrando resultados más certeros.

Las pruebas realizadas a la propuesta de solución permitieron detectar y corregir los errores presentes, obteniéndose un producto final con las funcionalidades requeridas.

Recomendaciones

Al finalizar la presente investigación se recomienda:

Implementar la interpretación de otros estándares de catalogación de metadatos, además de LOM, para una mejor integración de SISDREA con repositorios de RED.

Referencias bibliográficas

1. **Patricia Ibañez Carrasco, Gerrardo Garcia Torres.** *Informática/ Computer Science*. Pimera. México DF : Cengage Learning Editores, S.A, 2009. ISBN 10:607-481-095-5.
2. **Zapata, Martha.** *Recursos educativos digitales: conceptos básicos*. Universidad de Antioquia. 2012.
3. **Nestor Duque, Demetrio Ovalle, Julian Moreno.** *Objeto de aprendizaje, repositorio y federaciones, conocimiento para todos*. 2015.
4. **Neil Butcher, Asha Kanwar.** *Guía Básica de recursos educativos abiertos*. París : s.n., 2015. ISBN: 978-9-233000-20-9.
5. **Yoelkys Hernández Aracil, Romilio Anibal Corona Prendes.** *Sistema Basado en casos para la identificación de problemas de diseño en recursos educativos abiertos*. La habana, Cuba UCI : s.n., 2015.
6. **Prieto, Fernando Posada.** Diseño de recursos digitales educativos. [En línea] 27 de marzo de 2012. [Citado el: 19 de febrero de 2019.] <https://canaltic.com/blog/?p=889>.
7. **Bobenrieth, Manuel Astete.** Normas para revisión de artículos originales en Ciencias de la Salud. [En línea] 2002. [Citado el: 19 de enero de 2019.] http://www.aepc.es/ijchp/articulos_pdf/ijchp-54.pdf .
8. **Camoanario, J. M.** *El sistema de revisión por expertos (peer review): muchos problemas y pocas soluciones*. 2002.
9. **Roxana Cañizales González, Javier Soler Martín, Orlando Salvador Broche, Vivian Estrada Sentí.** *El Repositorio de Objeto de Aprendizaje RHODA desde su perspectiva colaborativa*. La Habana : Instituto de Información Científica y Tecnológica, 2013. ISSN: 0864-4659.
10. **Watson, Ian.** *Aplicaciones del razonamiento basado en casos sistema de técnicas empresariales*. 1997.
11. **Yadira Kiquey Ortiz Chow, Pedro Bañuelos Aguilar, Jorge Rodas-Osollo.** *Razonamiento Basado en Casos*. 2016.
12. **Salvat, Gros.** *Evolución y retos de la educación virtual: Construyendo el e-learning del siglo XXI*. Begoña : Editorial OC. p. 180, 2011.
13. **Rodríguez, Silvia Irene Adame.** *Instrumento para evaluar Recursos Educativos Digitales, LORI - AD*. 2015.
14. **Toll Palma, Yuniet del Carmen.** *La Calidad de los Objetos de Aprendizaje producidos en la Universidad de las Ciencias Informáticas*. . La habana : s.n., 2011. 1135-9250.
15. **García, José Molina y Jesús.** *Técnicas de Análisis de Datos*. 2006.
16. **Ruiz, Sergio.** El algoritmo K-NN y su importancia en el modelado de datos. [En línea] Analítica WEB , 20 de julio de 2017. [Citado el: 3 de junio de 2019.] <https://www.analiticaweb.es/algoritmo-knn-modelado-datos/>.
17. **González, Alejandro López y Viviana Maura.** *La técnica de ladov*. Buenos Aires, Argentina : s.n., 2003.
18. **Sanchez, Natalia Martinez.** *El Razonamiento Basado en Casos en el ámbito de la Enseñanza/Aprendizaje*. s.l. : Red de Revistas Científicas de América Latina y el Caribe, España y Portugal., 2008. ISSN: 1657-7663.
19. **Anibal Bregón, Arancha Simón, Carlos Alonso, Juan José Rodríguez, Belarmino Pulido, Isaac Moro.** *Un sistema de razonamiento basado en casos para la clasificación de fallos en sistemas dinámico*. 2005.
20. **Fabian Hernandez, Arellano Millward.** *El concepto de distancia y su Aplicación en Estadística Multivariada*. México, Brown : s.n., 2015.
21. **Carlos Eduardo Negri, Eduardo Luis De Vito.** *Introducción al razonamiento aproximado: lógica difusa*. Argentina : s.n., 2006. 1684-1859..

22. **Guzmán, Clara López.** *Los repositorios de objetos de aprendizaje como soporte a un entorno e-learning.* España Salamanca : s.n., 2005.
23. **González, María Luz Cacheiro.** *Recursos educativos TIC de informaciónn colaboración y aprendizaje.* Universidad de Sevilla España : Pixel-Bit. Revista de Medios y Educación, núm. 39, 2011.
24. **Jiménez, Fernando Jorge Mortera.** *Buenas prácticias para el uso académico de Recursos Educativos Abiertos y Objetos de Aprendizaje.* Monterrey, Costa Rica : s.n., 2013.
25. **EcuRed.** Repositorio de recursos educativos abiertos del MES. [En línea] [Citado el: 18 de octubre de 2018.] <http://revistas.mes.edu.cu>.
26. **Mila Gascó, Carlos Jiménez.** *Interoperability in the justice field: variables that affect implementation.* Slovenia : s.n., 2011.
27. **Jorge Ávila, Jorge Garcia.** ¿ Qué son los servicios Web y para que sirven? [En línea] 2016. [Citado el: 2019 de enero de 8.] <https://basicinfoweb.wordpress.com/2016/04/20/que-son-los-servicios-web-y-para-que-sirven/>.
28. **Tim Bray, Jean Paoli.** Extensible Markup Language (XML). [En línea] 2015. [Citado el: 2018 de noviembre de 3.] <http://www.renderx.com/~renderx/Demos/fo2html/xml.pdf>.
29. **Tim Bray, Jean Paoli.** *Extensible Markup Language(XML) 1.0.* 1998. 8879:1986.
30. **Fernandez, Oscar Belmonte.** *Introducción al lenguaje de programación Java: Una guía básica.* 2005.
31. **Chemical, Osmandi.** Qué es JavaFX? [En línea] [Citado el: 2018 de octubre de 3.] <https://medium.com/@osmandi/qu%C3%A9-es-javafx-41cfc327bdf2>.
32. **Krochmalski, Jarosław.** *IntelliJ IDEA Editions Comparison.* s.l. : Packt Publishing Ltd, 2014, 2014. ISBN: 1784398691.
33. **Sánchez, Natalia Martínez.** *El Razonamiento Basado en Casos en el ámbito de la Enseñanza/Aprendizaje.* *Revista de Informática Educativa y Medios Audiovisuales.* 2008.
34. **Ivar Jacobson, Grady Booch y James Rumbaugh.** *El proceso unificado de desarrollo de software.* 1999.
35. **Ticona.** Metodología SCRUM para el desarrollo de software y gestión de proyectos en las pequeñas y medianas empresas. Juliacan : s.n., 2014.
36. **Sintya Milena Meléndez, María Elizabeth Gaitan y Neldin Noel Pérez.** *Sistema web de evaluación al desempeño docente Unan-Managua empleando la metodología ágil de Programación Extrema.* Nicaragua : s.n., 2016.
37. **Scott., Ambler.** The Agile Unified Process (AUP). [En línea] [En línea], 2005. <http://www.ambysoft.com/unifiedprocess/agileUP.html>.
38. *Metodología de desarrollo para la Actividad productiva de la UCI.* **Sánchez, Tamara Rodríguez.** La Habana. Cuba : s.n.
39. **Laura Lozano, Javier Fernández.** *Razonamiento Basado en Casos: Una Visión General.* España : s.n., 2009.
40. **Corona Prendes, Frank David y Trujillo Irragorri, Luis Enrique.** *Sistema basado en casos para predecir la ocurrencia de reacciones adversas en la consulta médica.* La Habana Cuba UCI : s.n., 2013.
41. **Rivero., Elaine López.** *Sistema Basado en Casos para contribuir a la disminución de los costos de la calidad de los proyectos del Centro de Telemática.* La Habana, Cuba UCI : s.n., 2013.
42. **Morales, Gretel María Pérez.** *Sistema Basado en Casos para predecir la evaluación del estado de proyectos.* La Habana, Cuba UCI : s.n., 2014.

43. **Ofelia Peláez Soto, Yanelis Hierrezuelo Ramírez.** *Sistema basado en casos para el trabajo didáctico con estudiantes de bajo rendimiento académico en la asignatura Matemática Discreta I*. La Habana, Cuba UCI : s.n., 2016.
44. **Matos, Eliuvis Matos.** *Sistema Basado en el Conocimiento para asistir el diagnóstico y sugerir tratamientos en pacientes hemipléjicos con alteraciones motoras.* La Habana, Cuba UCI : s.n., 2016.
45. **Arletis Velázquez Ramírez, Leduan Bárbaro Rosell Acosta.** *Sistema inteligente para la resolución práctica de problemas de Inteligencia Artificial utilizando la programación lógica.* La Habana : s.n., 2016.
46. **Rafael Perez, Iliana Gutierrez, Telleria Rodríguez.** *Un SBC para la toma de decisiones en condiciones de incertidumbre.* . Santa Clara : s.n., 2002.
47. **Pressman.** *Ingeniería de software: Un enfoque práctico. 6ta ed.* Nueva York : EUA: McGrawHill, 2007.
48. **Sommerville.** *Software Engineering* . s.l. : Pearson Educación, 2006. ISBN 84782907.
49. **Guilherme Siqueira, Carlos Eduardo Vazquez.** *Ingeniería de requisitos: Software orientado al negocio.* 2015.
50. **Garzás, Javier.** La historia de usuario no es el “requisito” de las metodologías ágiles. [En línea] [Citado el: 6 de diciembre de 2018.] <http://www.javiargarzas.com/2011/12/historia-de-usuario-diferente-derequisito.html>..
51. **Anais González, Gabriel Urbina.** Arquitectura en tres niveles. [En línea] 2015. [Citado el: 2019 de abril de 28.] <https://basesdedatos2.wordpress.com/2015/06/25/arquitectura-de-base-de-datos-arquitectura-de-tres-niveles/>.
52. **Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides.** Elements of reusable object-oriented software. [En línea] [Citado el: 2019 de marzo de 5.]
53. **Larman, Craig.** *UML y patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado.* 2003. ISBN: 84-205-3438-2.
54. **Pressman, Roger.** *Ingeniería de Software. Un enfoque práctico.* . 2010.
55. **Franco, Andrés Fernando.** Diagrama paquetes, colaboracion y componetes. [En línea] 28 de enero de 2011. [Citado el: 6 de mayo de 2019.] <https://es.slideshare.net/andrescofran/diagrama-paquetes-colaboracion-y-componetes-6738524>.
56. **Eduardo Hernández Gómez, Angela Consuelo Romero.** Guía de estándares de desarrollo. [En línea] enero de 2015. [Citado el: 2019 de marzo de 5.] http://www.saludcapital.gov.co/Formatos%20Gua%20Desarrollo%20SDS/Estandares%20de%20Desarrollo%20de%20Software_SDS_v1.3.8.pdf.
57. **Acedo, Jose.** Estándares de nomenclatura: Snake Case, Kebab Case, Camel Case. [En línea] 1 de septiembre de 2017. [Citado el: 2019 de enero de 15.] <http://programacion.jias.es/2017/09/estandares-de-nomenclatura-snake-case-kebab-case-camel-case/>.
58. **Navas, Juan Sebastian Ramírez.** *Análisis sensorial. Pruebas orientadas al consumidor.* Colombia : Universidad del Valle, 2012. ISBN: 2017-6850.
59. **Yois Pascual, Cesar Omar Jaramillo Fredy Antonio Verástegui.** *Desarrollo de objetos virtuales de aprendizaje como estrategia para fomentar la permanencia estudiantil en la educación superior.* Bogotá : s.n., 2017.
60. **Antonio Bautista García, Carmen Alba Pastor.** ¿Qué es la tecnología educativa? Autores y significados. 1997, Vol. 4, págs. 51-62.

Referencias bibliográficas

61. **Yordi Chaveco Bustamante, Yoslenys Roque Hernandez y Jesus Vilches Pupo.** *Sistema web inteligente apoyado en mapas conceptuales para la asignatura Álgebra Lineal en la Universidad de las Ciencias Informáticas.* La Habana : s.n., 2013.
62. **Duque, Raul González.** Mundo geek. [En línea] 2015. [Citado el: 5 de marzo de 2019.] [http://mundogeek.net/archivos/2009/03/16/colecciones-en-java/..](http://mundogeek.net/archivos/2009/03/16/colecciones-en-java/)
63. **Fernando Alonso, Loic Martínez, Francisco Javier Segovia.** *Introducción a la ingeniería de software.* Zaragoza, España : s.n., 2005. ISBN: 84-96477-00-2.
64. **Rodríguez Sánchez, T.** *Metodología de desarrollo para la Actividad productiva de la UCI.* Universidad de las Ciencias Informáticas, La Habana, Cuba. : s.n., 2015.
65. **Peñuela, Manuel Cillero.** manuel.cillero.es. [En línea] [Citado el: marzo de 3 de 2019.] [https://manuel.cillero.es/doc/metrica-3/tecnicas/pruebas/integracion/.](https://manuel.cillero.es/doc/metrica-3/tecnicas/pruebas/integracion/)
66. **Sanchez, Tamara Rodríguez.** *Metodología de desarrollo para la actividad productiva de la UCI.* La Habana : s.n.
67. **Bregón, Anibal.** *Un sistema de razonamiento basado en casos para la clasificación de fallos en sistemas dinámicos.* 2012. 84-9732-449-8.
68. **Rodríguez, Nuria Gómez.** *Las Pruebas de Integración como Proceso de la Calidad del Software en el Ámbito de las Telecomunicaciones.* Madrid : s.n., 2015.

Anexos

Anexo 1: Categorías del instrumento de revisión de recursos educativos LORI con sus respectivos problemas de diseño.

Categorías	Problemas de diseño
Calidad de los contenidos	<ul style="list-style-type: none"> • El contenido es incorrecto. • El contenido presenta omisiones o sesgos. • El nivel de detalle es inadecuado. • Las presentaciones no refuerzan los puntos clave ni las ideas significativas. • La información muestra sesgos en la representación de grupos étnicos o culturas.
Adecuación de los objetivos de aprendizaje	<ul style="list-style-type: none"> • Los objetivos de aprendizaje no son evidentes. • No existe una relación entre los contenidos, las actividades y las evaluaciones. • Los objetivos no son apropiados para el perfil de alumnado al que se dirige.
Feedback (Retroalimentación) y Adaptabilidad	<ul style="list-style-type: none"> • No aporta feedback relativo a la calidad o corrección de las respuestas del alumnado. • No mantiene un registro de las respuestas o del estilo de aprendizaje del alumno/a de cara

	<p>a adaptar las siguientes presentaciones de contenido.</p> <ul style="list-style-type: none"> • No hay simulación o un conjunto de herramientas que se adapten en función de las respuestas del alumno/a.
Motivación	<ul style="list-style-type: none"> • El contenido no es relevante en la vida del alumnado. • Las actividades son demasiado fáciles o demasiado difíciles para percibir el interés del alumnado. • Las características del objeto que suponen captar el interés del alumno son una distracción que interfiere con el aprendizaje. • El objeto no varía en el tono, no hay muestras de humor o novedad. • La interacción del alumno/a con el objeto no presenta consecuencias interesantes.
Diseño y presentación	<ul style="list-style-type: none"> • La fuente seleccionada o el tamaño de la misma reduce notablemente la velocidad de lectura. • Cierta información que es necesaria para la comprensión del objeto de aprendizaje es ilegible. • La calidad del vídeo o audio es insuficiente para el aprendizaje.

	<ul style="list-style-type: none"> • La selección de colores, imágenes o sonidos interfieren con los objetivos de aprendizaje. • El diseño de la información provoca un sobreesfuerzo innecesario para el procesamiento de la misma. • No hay suficientes encabezados o no son significativos para el alumno/a.
Usabilidad	<ul style="list-style-type: none"> • No hay posibilidad de interactividad, solo hay texto y/o imágenes estáticas. • Varios enlaces o botones no funcionan. • Hay una demora excesiva en la navegación. • El funcionamiento de la interfaz no es intuitivo y tampoco hay instrucciones. • El funcionamiento de la interfaz es inconsistente e impredecible.
Accesibilidad	<ul style="list-style-type: none"> • Los videos no tienen subtítulos. • No hay transcripciones para los archivos de audio. • Faltan etiquetas en las imágenes. • La comprensión de los gráficos requiere que el alumno/a pueda percibir los colores.

<p>Reusabilidad</p>	<ul style="list-style-type: none"> • Se refiere al módulo, curso o docente para el que fue diseñado originalmente. • Su uso depende críticamente de recursos de aprendizaje específicos y externos. • Sólo puede ser utilizado por un grupo pequeño de alumnos/as con un nivel de conocimiento previo alto y específico.
<p>Cumplimiento de estándares</p>	<ul style="list-style-type: none"> • No se provee suficiente metadata o no está formateado de acuerdo al estándar de objetos de aprendizaje del IEEE. • Fallan todos los tests de cumplimiento de directrices del W3C y SCORM (directrices de accesibilidad no incluidas).

Anexo 2: según la consulta realiza a los expertos Yasirys Terry González, Reiman Alfonso Azcuy y Leduan B Rosell Acosta donde completaron un formulario con su criterio de similitud entre cada uno de los valores por rasgo predictor se obtuvo como resultados los que se muestran a continuación.

Estructura: colección (CL), mixta (MX), lineal (LN), jerárquica (JE), en red (ER), ramificada (RM), compartimentada (CM), atómica (AT).

Tabla 23: Nivel de similitud de la categoría estructura (elaboración propia)

	CL	MX	LN	JE	ER	RM	CM	AT
CL	100	75	50	50	50	50	50	50
MX	-	100	50	50	50	50	50	50
LN	-	-	100	0	0	0	0	0
JE	-	-	-	100	0	0	0	0

ER	-	-	-	-	100	0	0	0
RM	-	-	-	-	-	100	0	0
CM	-	-	-	-	-	-	100	0
AT	-	-	-	-	-	-	-	100

Nivel de agregación: 1,2,3,4

Tabla 24: Nivel de similitud de la categoría nivel de agregación (elaboración propia)

	1	2	3	4
1	100	75	50	25
2	-	100	75	50
3	-	-	100	75
4	-	-	-	100

Estado: Borrador, final, revisado, no disponible

Tabla 25: Nivel de similitud de la categoría estado (elaboración propia)

	Borrador	Final	Revisado	No disponible
Borrador	100	25	50	0
Final	-	100	50	0
Revisado	-	-	100	0
No disponible	-	-	-	100

Tipo de interacción: Activo, expositivo, mixto, indefinido.

Tabla 26: Nivel de similitud de la categoría tipo de interacción (elaboración propia)

	Activo	Expositivo	Mixto	Indefinido
--	--------	------------	-------	------------

Activo	100	0	50	25
Expositivo	-	100	50	25
Mixto	-	-	100	25
Indefinido	-	-	-	100

Tipo de recurso educativo: Ejercicio(Ej), simulación(Sim), cuestionario(Cues), diagrama(Dia), figura(Fig), gráfico(Gra), índice(in), diapositiva(Dp), tabla(tb), texto narrativo(Tn), examen(Ex), experimento(Exp), enunciado de problema (Ep), autoevaluación(Ae).

Tabla 27: Nivel de similitud de la categoría tipo de recurso educativo (elaboración propia)

	Ej	Si m	C ue s	Dia	Fig	Gra	In	Dp	Tb	Tn	Ex	Exp	E p	Ae
Ej	100	0	75	0	0	0	0	0	0	0	25	0	0	50
Sim	-	10 0	0	0	0	0	0	0	0	0	0	75	0	0
Cue s	-	-	10 0	0	0	0	0	0	0	0	50	0	2 5	25
Dia	-	-	-	10 0	75	75	0	0	0	0	0	0	0	0
Fig	-	-	-	-	10 0	75	0	0	0	0	0	0	0	0
Gra	-	-	-	-	-	100	0	0	0	0	0	0	0	0
In	-	-	-	-	-	-	100	0	75	0	0	0	0	0
Dp	-	-	-	-	-	-	-	100	0	0	0	0	0	0
Tb	-	-	-	-	-	-	-	-	100	0	0	0	0	0

Tn	-	-	-	-	-	-	-	-	-	100	50	0	75	25
Ex	-	-	-	-	-	-	-	-	-	-	100	25	50	75
Exp	-	-	-	-	-	-	-	-	-	-	-	100	0	50
Ep	-	-	-	-	-	-	-	-	-	-	-	-	100	0
Ae	-	-	75	-	-	-	-	-	-	-	-	-	-	100

Nivel de interacción = Densidad semántica: Muy bajo, bajo, medio, alto, muy alto.

Tabla 28: Nivel de similitud de las categorías nivel de interacción y densidad semántica (elaboración propia)

	Muy bajo	Bajo	Medio	Alto	Muy alto
Muy bajo	100	75	50	25	0
Bajo	-	100	75	50	25
Medio	-	-	100	75	50
Alto	-	-	-	100	75
Muy alto	-	-	-	-	100

Propósito: disciplina (1), idea (2), prerrequisito (3), objetivo educativo (4), restricciones de acceso (5), nivel educativo (6), nivel de destreza (7), nivel de seguridad (8).

Tabla 29: Nivel de similitud de la categoría propósito (elaboración propia)

	1	2	3	4	5	6	7	8
1	100	50	50	50	50	50	50	50

2	-	100	85	85	85	85	85	85
3	-	-	100	50	75	50	50	50
4	-	-	-	100	0	75	75	75
5	-	-	-	-	100	0	0	0
6	-	-	-	-	-	100	50	50
7	-	-	-	-	-	-	100	50
8	-	-	-	-	-	-	-	100

Anexo 3: historias de usuario

Número: 1	Nombre del requisito: insertar repositorio.
Programador: Adriana Valdés Avilés	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en Desarrollo: que no exista un repositorio para insertar.	Tiempo Real: 5 días
Descripción: Una vez que el usuario presione el botón gestionar Repositorio se mostrar la vista donde podrá insertar un nuevo repositorio de recursos educativos en el sistema rellorando los datos nombre y dirección, para posteriormente presionando el botón insertar se adicionará el repositorio a la tabla de repositorios y al XML del sistema.	
Observaciones:	
Prototipo de interfaz:	

Gestionar conexión con repositorios		
Nombre	Dirección (URL)	Seleccione
Biblioteca Escolar Digital	http://www.bibliotecaescolardigital.es	<input type="checkbox"/>
Buscador Infantil	http://www.buscadorinfantil.com	<input type="checkbox"/>
Educar	http://www.educarnet	<input type="checkbox"/>
Educacontic	http://www.educacontic.es	<input type="checkbox"/>
EducaRed	http://www.educared.net	<input type="checkbox"/>

Nombre:

Dirección:

Número: 2		Nombre del requisito: Modificar repositorio.	
Programador: Adriana Valdés Avilés		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 5 días	
Riesgo en Desarrollo: No se identificaron riesgos.		Tiempo Real: 5 días	
<p>Descripción: una vez que el usuario presione gestionar repositorios se mostrará una vista encargada de la gestión de los mismos, luego deberá importar el xml que contiene los datos de los repositorios. Al realizar la importación se mostrarán estos datos en una tabla, donde el usuario podrá seleccionar los datos de los repositorios que desea modificar y luego presionará el botón modificar para actualizar los cambios.</p>			
<p>Observaciones:</p> <p>los campos nombre y dirección no pueden estar vacíos</p>			

la URL tiene que tener la forma {http, https, ftp}://(...)

Prototipo de interfaz:

Gestionar conexión con repositorios		
Nombre	Dirección (URL)	Seleccione
Biblioteca Escolar Digital	http://www.bibliotecaescolardigital.es	<input type="checkbox"/>
Buscador Infantil	http://www.buscadorinfantil.com	<input type="checkbox"/>
Educar	http://www.educar.net	<input type="checkbox"/>
Educacontic	http://www.educacontic.es	<input type="checkbox"/>
Educared	http://www.educared.net	<input type="checkbox"/>

Nombre:

Dirección:

Número: 4	Nombre del requisito: Importar repositorio.
Programador: Adriana Valdés Avilés	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 4 días
Riesgo en Desarrollo: que no se encuentre la dirección donde se encuentra el XML con los repositorios	Tiempo Real: 4 días
Descripción: Para importar un listado de repositorios se debe seleccionar el botón importar y luego seleccionar en la ubicación donde se encuentra el XML con el listado de repositorios.	
Observaciones:	

Prototipo de interfaz:

Gestionar conexión con repositorios		
Nombre	Dirección (URL)	Seleccione
Biblioteca Escolar Digital	http://www.bibliotecaescolardigital.es	<input type="checkbox"/>
Buscador Infantil	http://www.buscadorinfantil.com	<input type="checkbox"/>
Educar	http://www.educarnet	<input type="checkbox"/>
Educacontic	http://www.educacontic.es	<input type="checkbox"/>
EducaRed	http://www.educared.net	<input type="checkbox"/>

Nombre:

Dirección:

Número: 5		Nombre del requisito: Exportar repositorio.	
Programador: Adriana Valdés Avilés		Iteración Asignada: 1	
Prioridad: Media		Tiempo Estimado: 4 días	
Riesgo en Desarrollo: No se identificaron riesgos.		Tiempo Real: 4 días	
<p>Descripción: Para exportar una lista de repositorio el usuario debe acceder a la vista gestionar repositorios dando clic en el botón gestionar repositorios y luego de tener los repositorios que se desean exportar con los datos correctos en la tabla del sistema, se debe presionar el botón exportar y elegir la dirección donde se desea almacenar dicho listado de repositorios en un XML.</p>			

Observaciones:

Prototipo de interfaz:

Gestionar conexión con repositorios		
Nombre	Dirección (URL)	Seleccione
Biblioteca Escolar Digital	http://www.bibliotecaescolardigital.es	<input type="checkbox"/>
Buscador Infantil	http://www.buscadorinfantil.com	<input type="checkbox"/>
Educator	http://www.educarnet	<input type="checkbox"/>
Educacontic	http://www.educacontic.es	<input type="checkbox"/>
EducaRed	http://www.educaRed.net	<input type="checkbox"/>

Nombre:

Dirección:

Número: 6	Nombre del requisito: Mostrar listado de repositorios.
Programador: Adriana Valdés Avilés	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 2 días
Riesgo en Desarrollo: No se identificaron riesgos.	Tiempo Real: 2 días
Descripción: Para poder ver el listado de repositorios el usuario debe acceder a la vista gestionar repositorios y cargar los repositorios a través de la funcionalidad importar repositorios.	
Observaciones:	
Prototipo de interfaz:	

Gestionar conexión con repositorios		
Nombre	Dirección (URL)	Selección
Biblioteca Escolar Digital	http://www.bibliotecaescolardigital.es	<input type="checkbox"/>
Buscador Infantil	http://www.buscadorinfantil.com	<input type="checkbox"/>
Educar	http://www.educarnet	<input type="checkbox"/>
Educacontic	http://www.educacontic.es	<input type="checkbox"/>
EducaRed	http://www.educaRed.net	<input type="checkbox"/>

Nombre:

Dirección:

Número: 7 **Nombre del requisito:** Solicitar recursos educativos a los repositorios seleccionados.

Programador: Adriana Valdés Avilés **Iteración Asignada:** 1

Prioridad: Alta **Tiempo Estimado:** 2 días

Riesgo en Desarrollo: que no exista conexión a internet. **Tiempo Real:** 2 días

Descripción: Para realizar la conexión con los repositorios el usuario debe haber seleccionado el botón revisar recurso educativo en la pantalla principal, luego rellenar todos los datos del recurso educativo que se pretende revisar, para finalmente luego de presionar el botón aceptar se seleccionan los repositorios que se utilizarán y posteriormente se realiza la conexión con cada uno de ellos para obtener sus recursos educativos y/o sus metadatos.

Observaciones:
Prototipo de interfaz:

Número: 8	Nombre del requisito: Mostrar información sobre los elementos que se tienen en cuenta para la detección de problemas de diseño en recursos educativos.
Programador: Adriana Valdés Avilés	Iteración Asignada: 2
Prioridad: media	Tiempo Estimado: 4 días
Riesgo en Desarrollo: no se tenga conexión a internet.	Tiempo Real: 4 días
Descripción: una vez obtenidos los posibles problemas de diseño que presenta un recurso educativo digital, el usuario tiene la opción de ver cual fueron los elementos que se tuvieron en cuenta para la revisión de dicho recurso educativo. Mostrándose los recursos educativos recuperados, el caso base, la similitud que existe entre ellos y la cantidad de RED analizados.	
Observaciones:	
Prototipo de interfaz:	

Módulo de explicación

Nombre del caso base:

Cantidad total de recursos analizados:

Nombre del RE	Similitud	Tipo de interactividad	Estado	Tipo de RE	Contexto	Dificultad
Ecured	Caso Base	Activo	Borrador	Ejercicio	Primaria	Fácil
Encarta	0.435	Activo	Final	Simulación	Secundaria	Diffcil
La Neurona	0.256	Mixto	Revisado	Cuestionario	Primaria	Fácil
Aprende Jugando	0.365	Expositivo	Borrador	Experimento	Secundaria	Diffcil

Número: 9	Nombre del requisito: mostrar ayuda
Programador: Adriana Valdés Avilés	Iteración Asignada: 2
Prioridad: media	Tiempo Estimado: 4 días
Riesgo en Desarrollo:	Tiempo Real: 4 días
Descripción: Para mostrar la ayuda el usuario tiene que presionar, en la interfaz principal, el botón ayuda.	
Observaciones:	
Prototipo de interfaz:	

Número: 10	Nombre del requisito: almacenar la solución	
Programador: Adriana Valdés Avilés	Iteración Asignada: 1	
Prioridad: media	Tiempo Estimado: 4 días	
Riesgo en Desarrollo:	Tiempo Real: 4 días	
Descripción: el sistema podrá almacenar una nueva solución una vez revisada si el valor de similitud entre el caso base y el RED más semejante es menor a un umbral definido.		
Observaciones:		
Prototipo de interfaz:		

Número: 11	Nombre del requisito: Evaluar experto	
Programador: Adriana Valdés Avilés	Iteración Asignada: 2	
Prioridad: baja	Tiempo Estimado: 10 días	
Riesgo en Desarrollo: No se identificaron riesgos.	Tiempo Real: 10 días	
Descripción: una vez el usuario introduzca los datos referentes al caso base, rellenando un formulario referente al mismo, presionará el botón evaluar experto y se mostrará una vista donde tendrá que rellenar un segundo formulario, el cual, esta definido por las categorías del instrumento LORI. De esta forma se almacena en la base de datos local del sistema un nuevo recurso educativo.		
Observaciones: No pueden existir campos vacíos.		
Prototipo de interfaz:		

Evaluar

Calidad de los contenidos	Regular ▼
Adecuación de los objetivos de aprendizaje	Regular ▼
Feedback (Retroalimentación) y Adaptabilidad	Mal ▼
Motivación	Regular ▼
Diseño y presentación	Mal ▼
Usabilidad	Mal ▼
Accesibilidad	No procede ▼
Reusabilidad	Regular ▼
Cumplimiento de estándares	Regular ▼

Número: 11	Nombre del requisito: recuperar los recursos educativos semejantes al caso base.
Programador: Adriana Valdés Avilés	Iteración Asignada: 2
Prioridad: baja	Tiempo Estimado: 15 días
Riesgo en Desarrollo: que no existan repositorios de donde obtener los metadatos de los recursos educativos.	Tiempo Real: 15 días
Descripción: el sistema obtendrá los recursos educativos más semejantes una vez que se conecte a los repositorios o a la base de datos local, y realice el proceso de comparación para determinar la similitud entre el caso base y cada uno de los recursos	

recursos educativos. Obteniendo una cantidad de recursos educativos definida por el usuario.

Observaciones:

Prototipo de interfaz:

Anexo 4: carta de aceptación