



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

**Centro Nacional de Investigación
y Desarrollo Tecnológico**

Tesis de Maestría

Reestructura de Recursos de Aprendizaje

**presentada por
L.I. Arelia Desiderio Patiño**

**como requisito para la obtención del grado de
Maestra en Ciencias de la Computación**

**Directora de tesis
Dra. Olivia Graciela Fragoso Díaz**

Cuernavaca, Morelos, México. Junio de 2025.



Educación
Secretaría de Educación Pública



CONACYT
NACIONAL DE MÉXICO



Centro Nacional de Investigación y Desarrollo Tecnológico
Departamento de Ciencias Computacionales

Cuernavaca, Mor., **22/Mayo/2025**

OFICIO No. DCC/146/2025

Asunto: Aceptación de documento de tesis
CENIDET-AC-004-M14-OFCIO

CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial de **ARELIA DESIDERIO PATIÑO** con número de control **M23CE040**, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado **"Reestructura de recursos de aprendizaje"** y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

ATENTAMENTE

Excelencia en Educación Tecnológica®
"Conocimiento y Tecnología al Servicio de México"

Dra. Olivia Graciela Fragozo Díaz
Directora de tesis

Dr. Juan Carlos Rojas Pérez
Revisor 1

Dra. Blanca Dina Valenzuela Robles
Revisor 2

C.E.p. Depto. Servicios Educativos
Expediente / Estudiante



Educación
Secretaría de Educación Pública



Excmo. Sr. Subdirector Académico



Centro Nacional de Investigación y Desarrollo Tecnológico
Subdirección Académica

Cuernavaca Mor., **20 mayo 2025**

Oficio No. SAC/113/2025

Asunto: Autorización de impresión de tesis

ARELIA DESIDERIO PATIÑO
CANDIDATA AL GRADO DE MAESTRA
EN CIENCIAS DE LA COMPUTACIÓN
PRESENTE

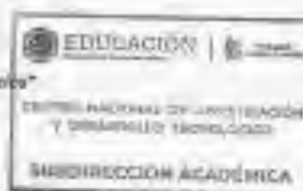
Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **"Reestructura de recursos de aprendizaje"**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE

Excelencia en Educación Tecnológica
"Cambiando y Revolucionando el Sistema de México"

CARLOS MANUEL ASTORGA-ZARAGOZA
SUBDIRECTOR ACADÉMICO



C.C., Departamento de Ciencias Computacionales
Departamento de Gestión Escolar

CMAZ/mg



2025
La Mujer Indígena

Interior Secretaría Pública CIL, CIL, Bando
C. P. 50460, Cuernavaca, Morelos 180, en (777) 667776 ext. 4341
e-mail: carlos.astorga@educacion.gob.mx / carlos.astorga@cei.gob.mx

cenidet



Dedicatoria

A mis padres, sin ellos esto no sería posible.

A mi familia, que son mi fortaleza.

A mis compañeros, por su invaluable apoyo.

A mis profesores, por brindarme sus conocimientos.

A mi comité revisor, por ser una fuente de inspiración.

A todos los que compartieron este camino conmigo y fueron cómplices de esta grata aventura.

Agradecimientos

Antes que nada, a Dios, por permitirme vivir esta experiencia.

A la Secretaría de Ciencias, Humanidades, Tecnología e Innovación (SECIHTI) por ayudarme a realizar este sueño brindándome el apoyo económico.

Al Tecnológico Nacional de México (TecNM), campus CENIDET, por permitirme ser parte de esta gran comunidad.

Con especial cariño a mi asesora, la Dra. Olivia Fragoso Díaz, por apoyarme y orientarme durante todo el proceso de esta investigación.

De igual manera extendo mi agradecimiento a la Dra. Blanca Dina Valenzuela Robles y al Dr. Juan Carlos Rojas Pérez por sus valiosas observaciones y aportaciones a este trabajo de investigación.

Al personal administrativo y al cuerpo docente de esta institución por las atenciones prestadas, especialmente a la Lic. Patricia Armas León, al Dr. René Santaolaya Salgado y al Dr. Máximo López Sánchez.

Y a todos los que me apoyaron en mi proceso de aprendizaje, mi más profundo agradecimiento.

Resumen

En la actualidad, y debido al aumento del uso de la tecnología, es común que bachilleratos, universidades y/o empresas recurran al uso de la educación a distancia, la cual día a día ha tomado mayor relevancia. Como consecuencia de este auge, es común encontrar recursos de aprendizaje de libre acceso en Internet. Sin embargo, se desconoce si cumplen con ciertos aspectos pedagógicos, de manera que garanticen una experiencia de aprendizaje efectiva (Aprendizaje virtual, 2014).

En este sentido, los recursos de aprendizaje abiertos deben estar diseñados para cumplir con metas educativas específicas, partiendo de la premisa de que éstos han sido evaluados por expertos y cumplen con estándares de calidad, de tal modo que se cumpla tanto con los objetivos planteados en programas académicos, como las necesidades de los estudiantes, en otras palabras, garantizar la calidad de los recursos de aprendizaje, así como el contenido y elementos que lo conforman (Aprendizaje virtual, 2014).

Bajo este escenario, un recurso de aprendizaje con mal diseño, que es poco claro o que está mal organizado, puede crear como consecuencia que futuros ingenieros de software no adquieran los conocimientos y/o competencias necesarias para adquirir las habilidades requeridas, es ahí donde surge la necesidad de este trabajo. Para fines de esta investigación se tratará con diagramas de clases UML usados como recursos de aprendizaje abiertos en formato PowerPoint, concretamente del área de Ingeniería de Software.

La solución propuesta consiste en un sistema que permite evaluar y reestructurar los defectos encontrados en un recurso de aprendizaje, los cuales se evalúan siguiendo las reglas del Modelo de Correctitud (Sandoval, 2023). Dicho modelo fue el resultado de un trabajo antecedente y el cuál es la base fundamental para el desarrollo de este trabajo de investigación.

Palabras clave: estética de diagramas de clases UML, legibilidad de diagramas de clases UML, atributos de calidad, métricas de diseño, olores de modelos, recursos de aprendizaje.

Abstract

Currently, due to the increasing use of technology, it is common for high schools, universities, and/or businesses to turn to distance education, which has become increasingly important. As a result of this boom, it is common to find freely accessible learning resources on the internet. However, it is unknown whether they meet certain pedagogical requirements to ensure an effective learning experience (Virtual Learning, 2014).

In this sense, open learning resources must be designed to meet specific educational objectives, based on the premise that they have been evaluated by experts and meet quality standards. This ensures that both the objectives set out in the academic programs and the needs of students are met; that is, ensuring the quality of the learning resources, as well as the content and elements that comprise them (Aprendizaje virtual, 2014).

In this scenario, a poorly designed, unclear, or poorly organized learning resource can prevent future software engineers from acquiring the knowledge or skills necessary to acquire the required competencies. Hence the need for this work. For this research, UML class diagrams will be used as open learning resources in PowerPoint format, specifically in the area of Software Engineering.

The proposed solution consists of a system that allows for the evaluation and restructuring of defects detected in a learning resource, which are assessed according to the rules of the Correction Model (Sandoval, 2023). This model was the result of previous work and constitutes the fundamental basis for the development of this research.

Keywords: Aesthetics of UML class diagrams, readability of UML class diagrams, quality attributes, design metrics, model smells, learning resources.

Contenido

I.	Introducción	2
I.1.	Descripción del problema	3
I.2.	Objetivos.....	3
I.3.	Alcances y limitaciones	4
I.4.	Organización del documento	4
II.	Marco conceptual	6
II.1.	Trabajos antecedentes.....	11
II.2.	Trabajos relacionados.....	14
II.3.	Análisis de trabajos relacionados	36
III.	Metodología de solución.....	43
III.1.	Modelo de Correctitud.....	46
III.2.	Requerimientos del sistema.....	49
III.3.	Análisis del sistema.....	50
III.4.	Diseño del sistema	58
III.5.	Implementación del sistema	64
III.6.	Casos de prueba	70
IV.	Análisis de resultados de pruebas	84
V.	Conclusiones	102
VI.	Referencias	105
	Anexo 1. Guía de usuario	113
	Anexo 2. Acrónimos.....	124
	Anexo 3. Resultados de evaluación	125
	Anexo 4. Productos de tesis.....	127

Índice de ilustraciones

Ilustración 1. Clase en OO.....	8
Ilustración 2. Atributos de legibilidad reestructurados en un recurso de aprendizaje.....	44
Ilustración 3. Porcentaje y Atributos de legibilidad evaluados en un recurso de aprendizaje.	44
Ilustración 4. Metodología de solución propuesta.....	45
Ilustración 5. Modelo de Correctitud de diagramas de clases UML (Sandoval, 2023)	47
Ilustración 6. Diagrama de Casos de uso ARE System	51
Ilustración 7. Diagrama de Estados de ARE System.....	59
Ilustración 8. Diagrama de Secuencia de ARE System.....	60
Ilustración 9. Diagrama de Secuencia de ARE System (Opción Upload)	60
Ilustración 10. Diagrama de Secuencia de ARE System (Opción Evaluar)	61
Ilustración 11. Diagrama de Secuencia de ARE System (Opción Procesar)	61
Ilustración 12. Diagrama de Secuencia de ARE System (Opción Download)	62
Ilustración 13. Diagrama de Clases de ARE System.....	63
Ilustración 14. Diagrama de Despliegue de ARE System	64
Ilustración 15. Ambiente de desarrollo de trabajo.....	65
Ilustración 16. Implementación de tecnologías web	65
Ilustración 17. Contenido del proyecto are.....	67
Ilustración 18. Área de trabajo MySQL Workbench.....	68
Ilustración 19. Servidor Tomcat en ejecución y sincronizado.....	69
Ilustración 20. Mensaje de confirmación de inicio de sesión exitoso.	71
Ilustración 21. Mensaje de inicio de sesión no exitoso.....	72
Ilustración 22. Mensaje de confirmación de registro de usuario exitoso.....	73
Ilustración 23. Formulario de registro de usuarios.....	74
Ilustración 24. Subida de archivos al sistema.....	75
Ilustración 25. Resultado del proceso de evaluación de un recurso de aprendizaje.....	77
Ilustración 26. Recurso de aprendizaje reestructurado.....	78
Ilustración 27. Archivo generado y listo para descargar.....	80
Ilustración 28. Confirmación de descarga del recurso de aprendizaje.....	81
Ilustración 29. Recurso de aprendizaje “154Loan.pptx”	85
Ilustración 30. Nivel de correctitud del recurso de aprendizaje.....	86
Ilustración 31. Recurso de aprendizaje reestructurado con la opción Text color.....	87
Ilustración 32. Nivel de correctitud después de cambiar el color del texto.....	87
Ilustración 33. Recurso de aprendizaje reestructurado con la opción Slide Background.....	88
Ilustración 34. Nivel de correctitud después de cambiar el color del panel de diapositivas.....	89
Ilustración 35. Recurso de aprendizaje reestructurado con la opción Shape Background.....	90
Ilustración 36. Nivel de correctitud después de cambiar el color del fondo de las clases.....	91
Ilustración 37. Recurso de aprendizaje reestructurado con la opción Font size.....	92
Ilustración 38. Nivel de correctitud después de cambiar el tamaño de fuente.....	93
Ilustración 39. Recurso de aprendizaje con grosor de líneas reestructurado	94
Ilustración 40. Nivel de correctitud final del recurso de aprendizaje “154Loan.pptx”	95
Ilustración 41. Resultados de evaluación, antes y después de usar ARE System.....	96
Ilustración 42. Resultados de la evaluación antes y después de usar ARE System	97

Ilustración 43. Resultados de la evaluación antes y después de usar ARE System	97
Ilustración 44. Resultados de la evaluación antes y después de usar ARE System	98
Ilustración 45. Resultados de la evaluación antes y después de usar ARE System	98
Ilustración 46. Resultados de la evaluación antes y después de usar ARE System	99
Ilustración 47. Resultados de la evaluación antes y después de usar ARE System	99
Ilustración 48. Resultados de la evaluación antes y después de usar ARE System	100
Ilustración 49. Resultados de la evaluación antes y después de usar ARE System	100
Ilustración 50. Número de incidencias de defectos de los diagramas de clases	101
Ilustración 51. Mapa de correlación entre los diferentes atributos de calidad	102
Ilustración 52. Página “Bienvenida”	114
Ilustración 53. Página Log In	115
Ilustración 54. Página Registrar nuevos usuarios.....	116
Ilustración 55. Mensajes de notificación de registro de nuevos usuarios.	117
Ilustración 56. Mensaje de ingreso fallido	118
Ilustración 57. Menú principal del sistema	118
Ilustración 58. Cargar archivos al sistema.	119
Ilustración 59. Confirmación de carga de archivos exitoso	120
Ilustración 60. Botón "Evaluar Recurso"	120
Ilustración 61. Mensaje de evaluación completada	121
Ilustración 62. Resultados de evaluación.....	121
Ilustración 63. Atributos para reestructurar.	122
Ilustración 64. Mensaje de confirmación de recurso reestructurado.	122
Ilustración 65. Botón "Descargar".....	123

Índice de tablas

Tabla 1. Trabajos relacionados con este tema de investigación.....	36
Tabla 2. Descripción de encabezados de la tabla de trabajos relacionados.....	41
Tabla 3. Equivalencia de atributos	48
Tabla 4. Ponderación de los atributos de calidad que evalúa ARE System	48
Tabla 5. Requerimientos Funcionales.....	49
Tabla 6. Descripción del Caso de Uso Iniciar Sesión	51
Tabla 7. Descripción del Caso de Uso Registrar Usuario	52
Tabla 8. Descripción del Caso de Uso Cargar Archivo.....	54
Tabla 9. Descripción del Caso de Uso Evaluar Recurso	54
Tabla 10. Descripción del Caso de Uso Procesar Recurso.....	55
Tabla 11. Descripción del Caso de Uso Generar Recurso	56
Tabla 12. Descripción del Caso de Uso Descargar Archivo.....	57
Tabla 13. Descripción del caso de prueba CP001 Iniciar sesión de usuario	70
Tabla 14. Descripción del caso de prueba CP001 Iniciar sesión de usuario (Fracaso)	71
Tabla 15. Descripción del caso de prueba CP002. Registrar usuario.....	72
Tabla 16. Descripción del caso de prueba CP002. Registrar usuario (Fracaso)	73
Tabla 17. Descripción del caso de prueba CP003. Cargar archivo.....	74
Tabla 18. Descripción del caso de prueba CP004 Evaluar recurso	76
Tabla 19. Descripción del caso de prueba CP005 Procesar recurso	77
Tabla 20. Descripción del caso de prueba CP006. Generar recurso	79
Tabla 21. Descripción del caso de prueba CP007. Descargar recurso	80
Tabla 22. Tabla comparativa de los valores de los atributos.....	95
Tabla 23. Número de incidencias	101
Tabla 24. Resultados de evaluación de ARESystem	125

Capítulo I

Introducción

Este capítulo describe el problema de investigación, se presentan los objetivos, así como de los alcances y limitantes de este tema de investigación, con el fin de introducir al lector en el contexto de lo relacionado a la reestructura de recursos de aprendizaje.

I. Introducción

El fácil acceso a internet ha permitido un incremento significativo en el uso de recursos de aprendizaje digitales. Tanto estudiantes como docentes recurren a este tipo de material, sin embargo, la frecuencia y la facilidad con que se accede a ellos ha llevado a la adopción de malas prácticas, siendo una de las más comunes el no verificar si los materiales que son consultados principalmente de internet y que son empleados como recursos de aprendizaje han sido elaborados con fines pedagógicos.

Como consecuencia de esto y debido a la gran cantidad de información que se crea cada momento, es común recurrir al uso de recursos digitales de libre acceso disponibles en internet, en ocasiones haciendo uso de alguna aplicación web como parte de la formación o capacitación sobre algún tema en específico. Al ser material que se encuentra disponible en internet de libre acceso, se desconoce si está diseñado adecuadamente o no, o si cumple con ciertos estándares de calidad, lo cual puede ocasionar que no se logren los objetivos de enseñanza.

Esto implica que los materiales disponibles deben seleccionarse objetivamente en función de las características del tema de interés (Educativa, 2015). Por ejemplo, el material que se produce con PowerPoint como es el caso de las presentaciones, son recursos que se emplean como material de apoyo para la clase con el objetivo de lograr asimilar el conocimiento fácilmente (Coscollá, 2009). Con la integración de este tipo de material, no sólo se busca facilitar la tarea del educador, sino también promover el autoaprendizaje en los estudiantes (Vásquez & Alexandra 2015).

Por ejemplo, en el área de Ingeniería de Software, es importante que en la fase de diseño de sistemas se cumplan los principios de diseño con el fin de minimizar el número de defectos en el software. Utilizar recursos de aprendizaje que no cumplen estos criterios puede provocar que no se adquieran los conocimientos necesarios y, en consecuencia, generar un software defectuoso. Por ello, es fundamental que los recursos de aprendizaje destinados a la formación de futuros ingenieros de software cumplan ciertos estándares con el fin de garantizar la adquisición de las competencias necesarias.

En este sentido, el uso de las herramientas de modelado UML, concretamente diagramas de clases, se basa en que pueden ayudar a mejorar la comprensión en los usuarios, lo que hace deseable la mejora en cuanto a la legibilidad en este tipo de diagramas, empleando principios de diseño que impactan en la claridad y el uso

efectivo de los diagramas, con el fin de mejorar la calidad y al mismo tiempo preservar la semántica.

De acuerdo con lo anterior, se propone una herramienta web que permite reestructurar de forma automatizada recursos de aprendizaje empleados en los procesos de enseñanza – aprendizaje en áreas afines del desarrollo de software, con el objetivo de mejorar aspectos enfocados a la legibilidad. En este documento, cuando se habla de recursos de aprendizaje, se hace referencia específicamente a diagramas de clases UML en formato PowerPoint tomados de internet.

El desarrollo de este trabajo se fundamenta en el Modelo de Correctitud propuesto en un trabajo antecedente (Sandoval, 2023), este modelo se usa como guía en el proceso de reestructuración, centrándose en aspectos relacionados con la calidad de recursos de aprendizaje.

I.1. Descripción del problema

Los diagramas de clases empleados como Recursos de Aprendizaje Abiertos son ampliamente utilizados en procesos de enseñanza aprendizaje en el área de Ingeniería de Software sin que se realice una evaluación de qué tan correctos están. Esta falta de evaluación puede ocasionar que no se logren los objetivos de aprendizaje y por lo tanto no se adquieran las competencias necesarias según los programas de estudios. Para apoyar el logro de las competencias requeridas, es necesario que los recursos de aprendizaje que se emplean sean lo más correctos posible.

I.2. Objetivos

Objetivo general:

Mejorar la calidad de los atributos de legibilidad de Recursos de Aprendizaje Abiertos en formato PowerPoint, concretamente diagramas de clases UML, fundamentado en las métricas de calidad definidas en un Modelo de Correctitud.

Objetivos específicos:

- Reestructurar recursos de aprendizaje, específicamente diagramas de clases UML en formato PowerPoint, asociados con criterios de legibilidad.
- Automatizar el proceso de reestructura de recursos de aprendizaje.
- Ser guía de apoyo en el proceso de desarrollo de recursos de aprendizaje.

- Evaluar la calidad de los recursos de aprendizaje.

I.3. Alcances y limitaciones

Para el propósito de este trabajo de investigación, se establecen los alcances y limitaciones descritos a continuación.

Alcances:

- Mejorar la calidad de recursos de aprendizaje asociados a la legibilidad.
- Emplear el Modelo de Correctitud de diagramas de clases UML (Sandoval, 2023).
- Los parámetros reestructurados corresponden al Dominio de Legibilidad.
- Se emplean lenguajes de programación como Java y HTML que permiten el desarrollo de servicios web.

Limitaciones:

- Un recurso de aprendizaje se limita a un diagrama de clases UML en una lámina de PowerPoint.
- El proceso de reestructura no atiende la parte semántica del recurso de aprendizaje.
- Debido a la magnitud de los escenarios planteados en el Modelo de Correctitud, no se atienden todos los atributos señalados.

I.4. Organización del documento

El Capítulo II contiene el Marco Conceptual en el que se describen algunos conceptos básicos y una breve descripción de los trabajos relacionados que proponen la mejora de diagramas de clases. El Capítulo III trata lo referente a la metodología de solución planteada para este tema de investigación y el proceso de desarrollo de la herramienta, según la metodología de desarrollo de sistemas. En el Capítulo IV se describe lo referente a las pruebas realizadas de acuerdo con el plan de pruebas establecido y los resultados obtenidos. En el Capítulo V se exponen las conclusiones del trabajo de investigación y los trabajos futuros.

Capítulo II

Marco conceptual

Este capítulo describe los conceptos básicos asociados a este trabajo de investigación, tratando de ofrecer una idea más clara de los términos asociados a este tema, así como de los trabajos relacionados que han abordado temáticas similares en cuanto a la mejora de diagramas de clases UML.

II. Marco conceptual

En seguida se presenta de manera breve una lista de los conceptos introductorios referentes a la reestructura de recursos de aprendizaje, especialmente aquellos asociados con diagramas de clases UML. También se incluye una breve descripción de los diferentes trabajos de investigación que han abordado temáticas relacionadas con este estudio.

1) Reestructurar

“Acción de reestructurar”, en este sentido, reestructurar es “modificar la estructura de una obra, disposición, empresa, proyecto, organización” (RAE, 2023).

2) Recurso de aprendizaje

Un recurso de aprendizaje hace referencia al material que se usa como guía en el proceso de enseñanza, y que sirve de apoyo para facilitar el proceso educativo en la comprensión de conceptos, habilidades o competencias específicas, con el fin de promover el dominio de los objetivos educativos en los estudiantes, fomentando un aprendizaje efectivo y significativo (UOC, 2015). Los recursos de aprendizaje incluyen textos, vídeos, plataformas educativas, presentaciones entre otros, y se pueden emplear en entornos formales como escuelas o de manera informal, como el autoaprendizaje.

3) Recursos educativos abiertos

Se considera como un Recurso Educativo Abierto (Open Educational Resource - OER) a los objetos de aprendizaje y materiales disponibles de libre acceso, relacionados con la educación o de investigación, con el objetivo de proporcionar una alternativa accesible a toda aquella persona que desee utilizarlos, y que generalmente aplica a recursos digitales, como cursos, libros de texto, tutoriales, conferencias y, en algunos casos, al software utilizado para hacer o distribuir material utilizado en la enseñanza, el aprendizaje o la investigación (reCrea, 2023). Por otro lado, en (Pinto, 2012), se define como “aquel material en formato digital que se ofrece de manera gratuita y libre para educadores, estudiantes y autodidactas para su uso en la enseñanza, el aprendizaje y la investigación”.

4) XML

De acuerdo con el W3C (World Wide Web Consortium), XML o Lenguaje de Marcas Extensible es una “herramienta independiente de software y hardware para almacenar y transportar datos de manera estructurada”. Algunas de las principales características de uso son, que permite una relación estructurada de los datos; permite el intercambio de información entre sistemas heterogéneos; permite crear etiquetas personalizadas; y

la simplicidad de la sintaxis. Lo que permite la interoperabilidad y la estandarización en la gestión de los datos entre aplicaciones (W3C, 2025).

5) UML

UML o Lenguaje de Modelado Unificado, es un lenguaje de modelado mediante gráficos usado para especificar, visualizar, construir y documentar los artefactos de un sistema de software (OMG, 2025). UML se encarga de captar la información sobre la estructura estática y el comportamiento dinámico del sistema, es decir, por un lado, definir los objetos importantes que formarán parte del sistema para su implementación, y por otro, la relación o comunicación que habrá entre éstos para cumplir los objetivos (Booch, 2004). Las técnicas de modelado UML facilitan la comprensión de ideas complejas y la comunicación entre miembros del equipo de desarrollo, tanto para el modelado de sistemas pequeños como para sistemas más complejos, ofreciendo una forma general y estandarizada de visualizar el diseño de un software, sin importar el lenguaje de programación.

6) PowerPoint

PowerPoint es una aplicación ofimática desarrollada por la empresa Microsoft, que a través de diapositivas proyectadas por medio de una computadora sirve de apoyo en presentaciones o exposiciones digitales de diversos temas. Con este tipo de aplicaciones es posible crear material que puede ser usado como apoyo en clase, a través de texto, gráficos, imágenes, animaciones, inserción de formas, cuadros de texto, colores, entre otros. Es una herramienta útil porque ayuda a visualizar ideas y conceptos de forma clara y atractiva, facilitando la comunicación y el entendimiento en el mundo empresarial, educativo o personal. Actualmente el formato que utiliza PowerPoint, se basa en el uso del formato Office Open XML (OOXML), un estándar basado en XML.

7) Apache POI

Apache POI (Poor Obfuscation Implementation) es una biblioteca de código abierto desarrollada en Java, proporcionada por la fundación Apache, que permite trabajar con archivos de Microsoft Office, incluyendo las presentaciones de PowerPoint. Esta biblioteca permite leer, escribir y manipular el contenido de archivos de Microsoft Office.

8) Objeto

Un objeto se define como una entidad con identidad, estado y comportamiento propio, del cual se toman las características individuales que comprenden y describen a una multitud de objetos. Una instancia de clase es otra manera para referirse a un objeto (Booch, 2004). Por su parte (Debrauwer, 2016) define a un objeto como una entidad

identificable y existente del mundo que puede ser abstracto o real, el cual posee un conjunto de atributos y métodos.

9) Clase

En el ámbito de la programación orientada a objetos, (Booch, 2004) describe a una clase como un conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones, comportamiento y representan un objeto dentro del sistema que se está modelando. En este sentido, una clase es una plantilla con características similares a partir de la cual se generan instancias (Blasco, 2019). Gráficamente, el símbolo que representa las clases consiste en una caja rectangular dividida en tres áreas: el área superior indica el nombre de la clase, el área central contiene los atributos, y el área inferior las acciones. La Ilustración 1 representa de manera general la notación de una clase en UML.

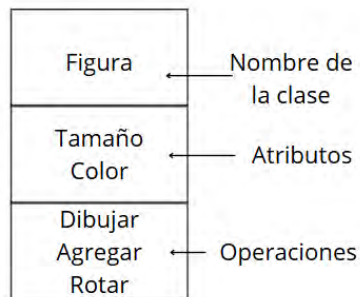


Ilustración 1. Clase en OO

10) Diagrama de clases

Un diagrama de clases es uno de los diagramas incluidos de acuerdo con la especificación oficial de UML dentro de la categoría de diagramas estructurales. Gráficamente está formado por rectángulos divididos en tres secciones (nombre de la clase, atributos y métodos) que representan las clases, conectados mediante líneas para indicar la manera en que estas se relacionan entre sí en un sistema informático (Kimmel, 2011). En ingeniería de software estos diagramas representan la vista estática de un sistema (Booch, 2004), es decir, no describen el comportamiento o la forma en la que interactúan los objetos (Kimmel, 2011). Su importancia en la programación orientada a objetos radica en la facilidad para transmitir ideas entre los ingenieros de sistemas, enfocándose en la estructura interna del software, es decir, la representación de los planos del sistema antes de su implementación.

11) Relación

Dentro de un diagrama de clases, las clases se conectan entre sí mediante líneas continuas (Kimmel, 2011) que representan las asociaciones entre clases, sin embargo, cada autor tiene su propia concepción sobre las relaciones, (Booch, 2004) dice que las

relaciones son la materialización de la conexión semántica entre los elementos del modelo, donde las relaciones se muestran como las líneas que conectan los rectángulos que representan las clases. El tipo de relación se distingue según el tipo de línea, su textura y por los adornos en sus extremos. Las relaciones entre clases pueden ser de herencia, agregación, asociación y composición.

12) Atributo

El término atributo se puede asociar a un conjunto de cosas reales o abstractas que constituyen alguna característica de interés (IEEE, 1997). O se puede considerar como la propiedad o característica inherente a una entidad que puede distinguirse cuantitativa o cualitativamente por medios humanos o automatizados, por ejemplo: personas, objetos, lugares, eventos, ideas (ISO/IEC/IEEE,2010). Los atributos según (IEEE, 1990) son las características de un elemento, como su color, tamaño o tipo de ese elemento. Entonces, en términos generales, un atributo es la propiedad compartida por los objetos de una clase (Teniente, 2004).

13) Métrica

De manera general, una métrica indica la medida cuantitativa que se usa para evaluar o monitorear un producto o proceso. En Ingeniería de Software se describe como la medida que cuantifica las características asociadas al proceso de desarrollo de software como la calidad, el rendimiento, incluso para medir la productividad del equipo de desarrollo (Fenton,1997). Otra de las definiciones para métrica dice que es una medida cuantitativa en el grado en que un sistema, componente o proceso posee un atributo dado (IEEE, 1990); considerando tanto la forma de medir o el método, como la escala de medición (Piattini, 2019).

14) Calidad

La calidad del software se define como el grado en que éste satisface los requisitos de usuario (ISO/IEC 25010). Esta puede ser en términos internos (atributos del software) o externos (comportamiento del software).

15) Legibilidad

En el diccionario legibilidad se define como “Que puede leerse. Especialmente, que puede leerse con facilidad” (RAE, 2023). En otras palabras, este término hace referencia a aquello que puede ser leído con facilidad, esto implica el hecho de usar un tamaño de fuente adecuado, que el contraste de colores entre el texto y el fondo sea adecuado, así como que el tipo de letra sea recomendable para una eficaz lectura.

16) Actividades de aprendizaje

Son aquellas acciones que un estudiante realiza con el fin de lograr adquirir ciertos objetivos de aprendizaje; con ello se busca que el aprendizaje sea más significativo y el cual se asocia a experiencias previas que el estudiante desarrolla en el ámbito educativo, las cuales le ayudarán a adquirir los conocimientos y habilidades de manera más relevante, de este modo promoviendo su aprendizaje (Glosario, 2008).

17) Educación abierta

Enfoque pedagógico de “filosofías y prácticas que buscan ampliar el acceso a la educación con fines de mayor equidad” (reCrea, 2023). Dicha equidad puede lograrse mediante la eliminación de ciertos requerimientos tales como requisitos de ingreso o costos, horarios de estudio, asistencia obligatoria a clases, y ritmos de avance o el uso libre de recursos y materiales didácticos (Glosario, 2008).

18) Competencias

En el ámbito de la educación, las competencias especifican el nivel de conocimientos y habilidades que una persona aplica ya sea para realizar actividades con éxito en el mundo laboral o en diferentes contextos (Glosario, 2008).

19) Competencias de aprendizaje

Son el conjunto de conocimientos y destrezas que debe poseer todo individuo, entre las más importantes según (Glosario, 2023) están: saber buscar, acceder, interpretar, organizar, sintetizar y comunicar, permitiéndole con base a los conocimientos resolver diferentes situaciones eficazmente.

20) Herramientas Didácticas o Material Didáctico

Son cualquier recurso o soporte que contenga mensajes didácticos, Por ejemplo: páginas web, guías de estudio, actividades, diapositivas, apuntes (Glosario, 2008).

21) Servicio web

Un servicio web es una aplicación que se puede describir, publicar, localizar e invocar a través de una red, el cual se basa en un conjunto de protocolos y estándares con el fin de intercambiar información. (Boulanger, 2015) lo define como un método disponible remotamente a través de una red, ya sea un intranet o Internet.

II.1. Trabajos antecedentes

1) *Uso de Reglas de Dominio para Evaluar Recursos de Servicios Web de Aprendizaje (Sandoval, 2023)*

(Sandoval, 2023) menciona que, el uso de recursos de aprendizaje electrónicos actualmente en la educación está ampliamente difundido. Las tecnologías actuales y la masificación del e-learning requieren que los recursos de aprendizaje que son utilizados de forma masiva sean evaluados y se mida su calidad, con el fin de que quien haga uso de estos recursos tengan información precisa sobre su calidad. La solución que se plantea consiste en medir la correctitud de los diagramas de clases UML contenidos en recursos de aprendizaje, por lo que se implementó un Modelo de Correctitud basado en una herramienta de software que a partir de un conjunto de reglas evalúa la calidad de los diagramas. El resultado es un reporte con los defectos encontrados. El Modelo de Correctitud que se plantea define como métricas la evaluación de la correctitud en el Dominio UML y la correctitud en Legibilidad.

2) *Systematic Review of Quality in Class Diagrams for Software Engineering Competencies (Fragoso et al., 2022)*

Se hace una revisión a diagramas de clases tomados de internet identificando los defectos más comunes y los atributos de calidad que deben cumplir de acuerdo con las métricas identificadas. También se habla de la importancia de la calidad de los recursos de aprendizaje ya que esto contribuirá a la adquisición de conocimientos en los usuarios. Al final se presentan los resultados de evaluar un conjunto de diagramas de clases que anteriormente pasaron por un filtro de selección. Los hallazgos encontrados fueron desde errores en la representación gráfica como el texto, el color y el trazado de las líneas, hasta errores que no cumplen con la formalidad del paradigma de programación orientada a objetos.

3) *QLERR (Quality Learning Resources Reader) un sistema para la medición de recursos de aprendizaje (Castillo et al., 2023)*

En este trabajo se presenta una herramienta para evaluar de manera automática recursos de aprendizaje en formatos digitales del tipo *.docx*, *.pdf* o *.pptx*. Dicha herramienta está constituida por una serie de módulos, uno de estos módulos se denomina “sistemas externos” que consta de un conjunto de medidores que individualmente evalúan el nivel de reusabilidad, legibilidad y correctitud, así como los subatributos de las métricas antes mencionadas. El medidor de correctitud califica diagramas UML incluidos en un archivo *.pptx* con base a un modelo de calidad.

Después de estudiar los defectos que se pueden encontrar en una muestra de diagramas UML, se establece un Modelo de Calidad con el cual se genera una herramienta que evalúa diagramas de clases empleados como recursos de aprendizaje. El siguiente paso se basa en un sistema que, a partir de los resultados obtenidos por dicha herramienta, reestructure diagramas de clases.

4) *Marco de servicios para medición de la calidad de formato de Recursos de Aprendizaje en Servicios Web (Pérez, 2021)*

En (Pérez, 2021) trata lo referente a métricas y atributos de calidad de recursos de aprendizaje respecto a la legibilidad del texto, el cual consiste en la evaluación de características físicas o de presentación del texto. Para fines de este trabajo de investigación, solo se consideran recursos de aprendizaje en formato de texto, teniendo como limitante la semántica de este. Para medir la legibilidad del texto se implementó una herramienta de software que mide la calidad de los recursos de aprendizaje y permite evaluar cuantitativamente dichos recursos. El sistema acepta archivos en formato .doc, .docx y pdf. Como resultado, la herramienta identifica las características físicas del documento, se indica el nombre de la métrica y si ésta cumplió o no con lo esperado. Algunos aspectos considerados son: alineación de texto, tipo de fuente, color de fuente, tamaño de la fuente, color de fondo, así como identificar errores tipográficos y de acentuación.

5) *Generador de Servicios Web de Aprendizaje Compuestos a partir de Recursos Educativos (Escobar, 2017)*

Debido al auge del uso de recursos de aprendizaje en línea, surge la necesidad de llevar más allá la funcionalidad de dichos recursos migrándolos a Servicios Web de Aprendizaje, con esto se busca aumentar la interoperabilidad y reusabilidad de dichos recursos. Este trabajo se centra en la creación de una herramienta de software capaz de integrar recursos de aprendizaje, denominada Generador de Servicios Web de Aprendizaje Compuestos. Los recursos incluidos son de tipo audio como *mp3*, video en formato *mp4*, imágenes de tipo *.png*, *.gif*, además de incluir documentos y páginas Web, por mencionar algunos. La limitante en este trabajo fue la capacidad de memoria, al crear recursos de aprendizaje se genera un desbordamiento de memoria que impide que puedan ser presentados.

6) Marco Orientado a Objetos para Medir la Calidad en Servicios Web de Aprendizaje (Salinas, 2017)

Se plantea la problemática del crecimiento y falta de “criterio” al usar recursos de aprendizaje de manera indiscriminada, también se menciona que los avances tecnológicos en la educación han cambiado la forma de enseñar, es decir, a medida que las exigencias de la educación cambian, también lo hacen los recursos de aprendizaje abiertos. Este tipo de material es empleado por un gran número de estudiantes y docentes, pero al usarlos sin un patrón o estándar en su diseño, se limita su reuso. Para dar solución a esta problemática, se plantea la opción de migrar los recursos de aprendizaje a servicios web, de este modo se disminuyen las restricciones de reusabilidad y portabilidad de los recursos de aprendizaje. Pero al cambiar estos recursos surge una nueva problemática, la de saber cuál es el grado de reusabilidad de los recursos de aprendizaje, y la de medir su calidad. Para ello se propone un modelo que mide la calidad de los servicios web. Los resultados obtenidos proporcionan información necesaria que ayuda a determinar la calidad de los servicios web de aprendizaje, ya que los datos recabados ofrecen información acerca del porcentaje de reusabilidad, además de proporcionar un resumen con los valores de los atributos que fueron tomados en cuenta para su evaluación.

7) Análisis y Definición de una Métrica para Evaluar la Legibilidad en Servicios Web de Aprendizaje (Rebolledo, 2017)

“Los objetos de aprendizaje pueden ser tratados como servicios web de aprendizaje”, pero al ser de fácil acceso, no se puede determinar la calidad de éstos, ni se dispone de algún método que ayude al usuario a seleccionar de manera objetiva aquellos que mejor se adecúen a sus necesidades. De ahí surge la necesidad de medir la calidad de los servicios web de aprendizaje, de modo que se permita la clasificación y al mismo tiempo, la selección de los recursos de aprendizaje. Como solución, primero se identifican los principales atributos de calidad de los recursos de aprendizaje; posteriormente se definió un conjunto de métricas con base en los criterios identificados. La solución plantea tanto la identificación de la calidad de los objetos de aprendizaje, como la identificación de los atributos de calidad, con el fin de medir la legibilidad de recursos de aprendizaje enfocados a documentos y presentaciones. También aborda cuestiones de legibilidad en servicios web de aprendizaje en formato de textos y presentaciones. Los criterios que se tomaron en cuenta para el caso de recursos de aprendizaje tipo texto son: longitud de palabra, tamaño de fuente, tipo de fuente; para el caso de presentaciones, se consideraron atributos como animación, audio, video, palabras por diapositiva, tamaño de fuente, entre otros. Se concluye que en un futuro el tema se podría implementar de manera automatizada.

II.2. Trabajos relacionados

Los trabajos relacionados descritos a continuación se clasifican según el enfoque o la metodología que usan para mejorar la legibilidad en diagramas de clases UML, procurando que en cada categoría se encuentren los trabajos con metodologías similares. La primera categoría trata aspectos relacionados con la legibilidad de recursos de aprendizaje; la siguiente se enfoca en lo relacionado al uso de lenguajes formales; en otra se presentan los trabajos que consideran el uso de modelos como guías; otra clasificación incluye los trabajos que emplean algoritmos de aprendizaje automático; por último, se mencionan aquellos trabajos que realizan estudios empíricos o que usan métricas de diseño orientadas a objetos. Sin embargo, debido a la variedad de técnicas y métodos que atienden la mejora de factores de diseño y estética de los gráficos de diagramas UML, esta clasificación puede variar según el punto de vista de cada lector, ya que cada metodología incluso puede combinar diferentes enfoques. A continuación, se presenta una breve descripción de los trabajos encontrados en la literatura.

I. Trabajos que tratan sobre la legibilidad en recursos de aprendizaje digitales.

1) Los recursos didácticos como apoyo en el proceso de enseñanza-aprendizaje de los estudiantes (Napa, 2023)

Hoy en día es común recurrir al uso de recursos de aprendizaje digitales, especialmente en el uso de presentaciones, ya que estas permiten presentar la información en diferentes formatos (p.ej., texto, audio, imágenes) y de manera atractiva, como material de apoyo para facilitar la adquisición del conocimiento. Este trabajo sugiere el uso de recursos de aprendizaje innovadores, pues esto ayuda al estudiante a asimilar el conocimiento de manera clara y atractiva, en él se mencionan aspectos referentes a los recursos de aprendizaje, como las ventajas de usar recursos de aprendizaje, el papel que juegan en el proceso de enseñanza y la importancia de éstos. El trabajo se basa en el desarrollo de un estudio estadístico a un grupo de estudiantes, el resultado fue una guía que ayude al docente en el manejo de los recursos de aprendizaje y conocer su desempeño en clase.

2) Buenas prácticas para crear presentaciones accesibles (Rajngewerc, 2022)

Este trabajo ofrece sugerencias con el fin de que las presentaciones sean claras al momento de exponer, de manera que el mensaje llegue a un mayor número de personas. Para lograr esto se recomienda entre otras cosas, conocer a la audiencia, hacer reuniones previas de manera que se puedan hacer correcciones antes a la

exposición, así como una serie de sugerencias para llevar a cabo una buena presentación. Además de estos consejos, también se dan algunas recomendaciones en cuanto al contenido de las diapositivas, esto incluye entre otros aspectos, la tipografía, el uso de imágenes o vídeos dentro de la presentación, así como el uso adecuado de los colores. Al final se ofrece una serie de consejos para crear presentaciones legibles y accesibles.

3) *Elementos para el desarrollo de presentaciones académicas en PowerPoint (Flores, 2012)*

Se presentan varias directrices tanto para el desarrollo de una adecuada presentación, como del diseño gráfico de las diapositivas, resaltando la importancia de la legibilidad, tanto de las diapositivas en aspectos asociados a la estructura del contenido y elementos a considerar durante la exposición.

4) *Aprendizaje de Word y PowerPoint en estudiantes de secundaria (Alarcón et. al, 2024)*

En un mundo donde la tecnología está en todos lados, los autores hablan sobre la importancia en el uso de herramientas como Word y PowerPoint, pues ayudan a mejorar en el estudiante sus habilidades digitales, su calidad educativa, y a fomentar nuevas formas de aprendizaje. Esta investigación lleva a cabo un estudio donde a los participantes se les pide realizar una serie de actividades en Word y PowerPoint, derivado de los resultados obtenidos, se determinó que el uso de un equipo de cómputo es indispensable, ya sea dentro de un ambiente escolar o en un ambiente laboral, ya que las competencias digitales juegan un papel muy importante, por lo que se concluye que la tecnología es crucial en el proceso de enseñanza-aprendizaje, pues ayuda a mejorar las capacidades de aprendizaje en los estudiantes.

5) *Evaluación y Mejora de la Calidad de la Presentación de Diapositivas como Herramienta Docente Universitaria (Garrido, 2017)*

En este trabajo se pide a un grupo de estudiantes universitarios que realicen las evaluaciones de las presentaciones de ciertas asignaturas. De acuerdo con los resultados obtenidos, las autoras plantean como la calidad de los elementos tales como la tipografía, el tamaño del texto, el uso de colores o la organización espacial de los elementos de una diapositiva se relacionan con el aprendizaje, lo que lleva a fomentar el buen uso de los espacios en las diapositivas. Esto se determina de acuerdo con ciertos principios, los cuales se listan dentro del artículo como son: principio de relevancia, de conocimiento, de discriminación y de organización perceptual, entre

otros. El objetivo se basa en “comprobar que las diapositivas funcionan para quienes tienen que funcionar”. Dicho de otro modo, determinar si las diapositivas resultan útiles para los estudiantes, ayudando en la asimilación de conceptos. El estudio permitió concluir que el tamaño del texto y la organización espacial de los elementos dentro de la diapositiva fueron los elementos con mayor relevancia.

6) *Los recursos educativos electrónicos: perspectivas y herramientas de evaluación (Pinto et. al, 2012)*

Debido a la variedad de recursos educativos ya sea en el formato y plataformas utilizadas o en el ámbito educativo, esta diversidad constituye una dificultad en la valoración de la calidad, tanto en el contenido, en la navegabilidad y en su ergonomía. Para ello se presenta EVALUAREED, un instrumento para la medición de recursos educativos digitales, así como un análisis de las diferentes perspectivas de evaluación. Esta herramienta se basa en un checklist de nueve criterios como son: calidad del contenido; objetivos y metas de aprendizaje; feedback; accesibilidad; requerimientos técnicos y efectividad del recurso desde el punto de vista del aprendizaje, entre otros. Estos indicadores definen las características necesarias para que un recurso de aprendizaje se considere adecuado desde el punto de vista pedagógico.

7) *Descripción de criterios sobre presentaciones digitales en PowerPoint con estudiantes del curso postuniversitario en competencias docentes (Hurtado, 2023)*

El objetivo de este trabajo se trata en describir los criterios utilizados para elaborar presentaciones digitales en PowerPoint. El cual consiste en dos estudios, el primero fue una prueba con la escala de Likert y para el otro se consideró una lista de cotejo. El primer método fue útil para medir características pedagógicas, de diseño y tecnológicas, en el segundo se miden los criterios considerados en la presentación, así como si el contenido es adecuado para el alumno. Algunos aspectos por considerar a la hora de elaborar presentaciones electrónicas son el texto, la calidad de la imagen y el formato. Este estudio permitió determinar que contar con guías como material de apoyo para la creación de presentaciones digitales favorece el proceso de enseñanza-aprendizaje. En conclusión, las presentaciones deben cumplir con ciertos parámetros que favorezcan la comprensión del tema.

8) *Guía para la evaluación y adecuación de materiales digitales accesibles (López et. al, 2021)*

Este artículo se basa sobre la revisión a la accesibilidad de materiales educativos digitales en la página web de la UJAT para estudiantes con deficiencias visuales, la cual tiene como objetivo proponer una guía para facilitar la accesibilidad en herramientas digitales tales como Word, PowerPoint y PDF. El proceso consiste en realizar una verificación manual mediante listas de cotejo y otra automática que ayude a detectar problemas de accesibilidad (p.ej. error, advertencia y sugerencia). El resultado de la comprobación indica las partes a corregir, con ello se busca garantizar el acceso a la educación tanto para aquellas personas con discapacidad visual como para personas sin discapacidad, permitiendo a los usuarios estudiar en condiciones iguales. Donde un documento accesible se considera como de mejor calidad, cuyo contenido es más comprensible para los usuarios. Por otro lado, en cuanto a los documentos busca garantizar el contenido de éstos, de manera que puedan ser leídos fácilmente por más personas. En conclusión, contar con una guía ayuda a crear documentos digitales accesibles, si bien su elaboración no supone un esfuerzo extra, asume un compromiso para ir avanzando en el proceso de inclusión, sin embargo, también se habla sobre una capacitación para la adecuación de las técnicas de enseñanza en el aula.

9) *Automated Grading of PowerPoint Presentations Using Latent Semantic Analysis (Borade et al., 2022)*

El trabajo del evaluador es tediosa y estresante cuando el proceso es manual, además de la falta de uniformidad en el proceso de evaluación, lo que hace necesario un proceso de evaluación objetiva. Por lo que este trabajo propone una técnica para la evaluación automática en las presentaciones en PowerPoint de estudiantes usando Análisis Semántico Latente (LSA), el objetivo es evaluar la calidad de la presentación en función de la presentación, la precisión y la autenticidad del contenido. La herramienta propuesta esta desarrollada en Python y se basa en el uso de algoritmos de aprendizaje automático.

10) *Visual Clarity Analysis and Improvement Support for Presentation Slides (Oyama, 2019)*

Las diapositivas se han vuelto material de apoyo que facilitan la transmisión de la información en diversos campos. Sin embargo, se carece de métodos de evaluación que puedan indicar si las presentaciones digitales son fáciles o no de comprender. Ya que la calidad de las presentaciones tiene una influencia directa en la comprensión de

éstas sobre la audiencia, se considera que las imágenes y palabras en una diapositiva ayudan a las personas a aprender de manera efectiva. Por otro lado, se considera que, para retener efectivamente la información presentada, se requiere que las diapositivas reduzcan el uso de texto, colores y objetos. Para este estudio se extrajeron las características de imagen, diseño y de contenido de 1000 diapositivas de una sola lamina elaboradas en PowerPoint. El método empleado se basó en la implementación de algoritmos de aprendizaje automático que ayudaron a predecir la claridad visual de las diapositivas, donde las características visuales y de estructura, fueron las que alcanzaron un 90% de precisión.

11) The Effect of PowerPoint on Reading Comprehension Improvement among High school students: A case Study in the City of Shoush (Holakopour, 2014)

El estudio consiste en un análisis estadístico con el fin de mostrar el impacto del uso de PowerPoint en la comprensión lectora de un grupo de estudiantes de nivel secundaria. Para ello el estudio se basó en unas pruebas antes y después de realizar un curso de capacitación y de este modo calcular las puntuaciones. Los participantes se asignaron aleatoriamente a los grupos de control y de experimentación. El análisis se basa en investigar la diferencia en la comprensión lectora entre un grupo que usa PowerPoint y los que usan el método tradicional. Los resultados indicaron que, si existe una diferencia significativa entre los estudiantes que usan PowerPoint y los que no, es decir, que el grupo que trabajo con material como PowerPoint tuvo mejor rendimiento, por lo que se puede concluir que el material en PowerPoint tiene un efecto significativo en la comprensión lectora. Lo que sugiere que el uso de material como PowerPoint puede mejorar el proceso de aprendizaje en los estudiantes. Una de las limitantes en este estudio fue el número reducido de participantes, así como las limitantes en el número de equipos para proyectar a los estudiantes.

Estos estudios enfatizan colectivamente la importancia de desarrollar recursos de aprendizaje legibles, demostrando como esto influye directamente en la comprensibilidad del usuario (Pinto, 2012). Sin embargo, estos trabajos se mencionan porque abordan aspectos relacionados con la legibilidad de los recursos de aprendizaje y cómo ésta puede afectar el proceso de aprendizaje, no obstante, se descartan como trabajos relacionados porque no están enfocadas a la legibilidad de diagramas de clases UML.

II. Trabajos relacionados enfocados a la implementación de lenguajes formales.

1) *Using Examples for Teaching Software Design: An Experiment Using a Repository of UML Class Diagrams (Karasneh, 2015)*

Este estudio estadístico se centra en evaluar los métodos de diseño de software aplicados en la enseñanza. Para ello, se llevó a cabo un experimento con dos grupos de estudiantes del área de ingeniería de software, a los cuales se les asignó una tarea de modelado. Al primer grupo se le proporcionaron ejemplos de diseños de software representados mediante diagramas de clases UML, mientras que al segundo grupo no. Los diagramas realizados por ambos grupos fueron evaluados por expertos en cuanto a comprensibilidad, diseño, extensibilidad e integridad; y de acuerdo con los resultados obtenidos, se concluyó que el uso de ejemplos produce en los estudiantes mejores diseños de software. A los estudiantes no se les indicó a qué grupo pertenecían.

2) *Evolutionary layout of UML class diagrams (Gudenberg, 2006)*

Se presenta un algoritmo evolutivo que diseña diagramas de clases. Un diagrama se representa como un nodo con aristas mediante coordenadas x e y. El algoritmo se basa en mutar las posiciones de los símbolos de las clases y las conexiones, guiado por una función de aptitud que evalúa la estética del diagrama. La función de aptitud combina nueve métricas de diseño, como la superposición de nodos, cruces de aristas, y la orientación de las relaciones jerárquicas. El algoritmo mostró éxito en la creación de diagramas de clases UML, pero sugiere aumentar el peso de métricas específicas para diagramas más grandes, como el caso de cruces de bordes o dar más énfasis en las jerarquías.

3) *UML diagram refinement (focusing on class- and use case diagrams) (Faitelson, 2017)*

Los diagramas UML de gran tamaño restan utilidad porque son difíciles de comprender, sin embargo, como solución se plantea refinar diagramas a diferentes niveles de abstracción. El problema es garantizar la consistencia de los diagramas a esos niveles de abstracción, para ello se define un conjunto de reglas de refinamiento de diagramas. El proceso consiste en una serie de transformaciones matemáticas, fundamentado en la teoría de homomorfismos de grafos, con el fin de construir modelos detallados a partir de abstracciones, sin alterar su estructura. La desventaja es que UML no define reglas que garanticen la eficiencia del método.

4) *Integrating and Refining UML Models (Liu et al., 2004)*

Utilizando el lenguaje formal OOL (Object-Oriented Language) se presenta un enfoque para integrar y refinar modelos UML. Este método busca abordar problemas de consistencia y refinamiento en modelos UML multivista, permitiendo integrar modelos estructurales y de comportamiento, con el fin de realizar transformaciones precisas y consistentes durante el desarrollo de sistemas. La metodología se basa en la formalización, desarrollando un conjunto de leyes de refinamiento de modelos UML que capturan los principios, la naturaleza y los patrones del diseño orientado a objetos. Con esto se busca asegurar que no se generen inconsistencias, es decir, el refinamiento de modelos UML en el proceso de desarrollo de sistemas, debe garantizar la consistencia y la futura evolución del sistema.

5) *Rule Modeling and Markup (Wagner, 2005)*

Este artículo propone modelar reglas para sistemas basados en UML integrando conceptos en lenguaje formal. Se menciona el uso de UML para representar vocabularios y reglas en una forma más visual y accesible. Por un lado, UML es ampliamente utilizado, pero carece de semántica formal, y por otro, los lenguajes semánticos formales no disponen de una notación visual. Por ello, se propone complementar UML con el Lenguaje de Restricciones de Objetos (OCL), para obtener un lenguaje general de modelado de reglas.

6) *Towards the Mechanized Semantics and Refinement of UML Class Diagrams (Sheng, 2019)*

“COQ: es un demostrador de teoremas interactivo lanzado por primera vez en 1989”. Este permite expresar mediante enunciados notaciones matemáticas, verificarlos mecánicamente y ayuda a encontrar pruebas formales para luego extraer un programa certificado de su especificación formal. Debido a que los diagramas UML carecen de una semántica formal, en este trabajo se emplean lenguajes matemáticos para formalizar la sintaxis y semántica de diagramas UML, específicamente diagramas de clases y objetos. Este enfoque propone el uso de lenguajes matemáticos a través del refinamiento de modelos, con el objetivo de definirlos formalmente. El proceso de refinamiento asegura que los modelos propuestos generen modelos más detallados, garantizando que sean consistentes con el modelo original. La verificación formal se lleva a cabo con el asistente COQ. La metodología empleada consiste en generar a partir del modelo UML un archivo XML, posteriormente este se transforma a lenguaje formal.

7) *Refactoring Class Hierarchies with KAVA (Streckenbach, 2004)*

Se propone KABA como un sistema que refactoriza jerarquías de clases java y se basa en análisis dinámico y estático, utilizando un lenguaje formal de especificación orientado a objetos (OOL, Object-Oriented Language) asegurando la coherencia entre diferentes niveles de abstracción. El enfoque estático se encarga de preservar el comportamiento del programa, por otro lado, la versión dinámica, busca garantizar el comportamiento del programa durante la ejecución de este. La implementación de esta investigación se basa en el uso del algoritmo Snelting/Tip. Snelting/Tip es una herramienta que analiza qué se puede hacer sin alterar el comportamiento.

8) *Overview of Slicing and Feedback Techniques for Efficient Verification of UML/OCL Class Diagrams (Shaikh, 2018)*

Se presenta la descripción de dos técnicas de corte disjuntas y no disjuntas para dividir diagramas de clases UML en submodelos, de modo que se pueda reducir la complejidad de los grandes modelos utilizando un enfoque de corte llamado Técnica de corte UML/OCL (UOST). Luego, estos submodelos se verifican por separado mediante alguna herramienta o motor de verificación formal. También se propone una técnica de retroalimentación para identificar submódulos infractores que no cumplen las restricciones de integridad en un diagrama de clases UML. Se concluye que la técnica de dividir en submodelos es eficiente y ofrece a los diseñadores de software verificar diagramas de clases complejos ayudando a centrar su atención a los submodelos incorrectos.

9) *On the impact of layout Quality to Understanding UML Diagrams: Size Matters (Störrle, 2014)*

De acuerdo con el autor, “la calidad del diseño, sin tomar en cuenta el tipo de diagrama, afecta la comprensión de diagramas UML”. El diseño juega un papel muy importante a la hora del desarrollo de software, la calidad de éste también está determinada por el nivel de experiencia del modelador. Este trabajo se basa en determinar el nivel de correlación entre la calidad del diseño y la comprensión del diagrama UML, basándose en el tamaño del diagrama, para lo cual se proponen medidas para el tamaño de diagramas UML. Esto permite concluir que una buena calidad en el diseño es particularmente útil en diagramas grandes, así como para modeladores con poca experiencia.

10) *A visual Base Framework for the Model Refactoring Techniques (Stolc, 2010)*

Este trabajo se centra en la refactorización de modelos de diagramas de clases UML considerándola como parte implícita en el ciclo de vida de desarrollo del software de

métodos ágiles. Primero se identifican los defectos en el modelo usando consultas OCL, estos se corrigen y transforman usando Lenguajes de Transformación basados en QVT (Query/View/Transformation). El resultado de la consulta se toma como entrada al proceso de transformación, el cual representa el script corregido.

11) *UML Model Refactoring as Refinement: A Coalgebraic Perspective (Barbosa, 2008)*

El Lenguaje Unificado de Modelado es una herramienta de apoyo en la representación de modelos en la creación de sistemas de software, sin embargo, carece de una definición semántica. Este trabajo se basa en la refactorización de modelos, la cual propone una base formal para la descripción, análisis y transformación de modelos. El proceso de refinamiento busca asegurar la equivalencia entre modelos durante el proceso de refactorización.

12) *CUCKOO: Una plataforma web para la verificación de modelos UML (Caballé et al., 2009)*

Se presenta una herramienta web que verifica de manera formal diagramas UML, para ello el modelo UML, las restricciones OCL y las propiedades de corrección se traducen en un modelo de satisfacción formal. Entonces, como solución, se propone CUCKOO, una plataforma web que de manera autónoma permita evaluar el trabajo de los estudiantes, esto con el objetivo de ayudar en su proceso de aprendizaje. La plataforma CUCKOO se basa en el conjunto de reglas textuales descritas en el lenguaje OCL. La herramienta contiene una lista predefinida de propiedades a verificar, clasificadas en aspectos como propiedades del modelo, propiedades de restricciones y propiedades de operación, y para cada ejercicio, el profesor define qué propiedades deben comprobarse.

13) *Refactoring OCL annotated UML class diagrams (Marković, 2007)*

La refactorización de diagramas de clases se asocia al aspecto visual, pero muchas veces se descuida la parte semántica. Este trabajo se basa en la formalización de reglas de refactorización de diagramas de clases UML y las clasifica según el tipo de restricción OCL que afecta, identificando cinco grupos Rename, PullUp, PushDown, Extract, Move. Las reglas de refactorización se formalizan usando modelos QVT (Query/View/Transformation), lo cual permite demostrar que las reglas de refactorización preservan la sintaxis y la semántica de los modelos a los que se aplican.

14) Formal Refactoring for UML Class Diagrams (Massoni, 2005)

La evolución del software es necesaria en el contexto del mundo real, sin embargo, la estructura definida originalmente no suele dar cabida a adaptaciones, por lo que se requiere de un proceso de reestructuración que a la vez garantice la parte semántica. Este trabajo presenta un enfoque formal para la refactorización de diagramas de clases UML empleando el lenguaje OCL para preservar la semántica formal del diagrama de clases. Los diagramas de clases UML junto con las restricciones definidas en OCL se traducen al lenguaje Alloy como método de comprobación para garantizar que las transformaciones mantienen la semántica formal.

15) UML Model Refactoring with Emphasis on Behavior Preservation (Hosseini, 2008)

Con la refactorización de diagramas UML se busca la mejora de estos, poniendo énfasis en no alterar su comportamiento, para ello se hace uso de lenguajes formales como métodos de verificación de las reglas de refactorización de los modelos. Los pasos de la refactorización se implementan utilizando la herramienta de transformación de gráficos AGG (Graph Transformation Tool) y para demostrar que la técnica de refactorización preserva el comportamiento original del modelo se usa el lenguaje ROOL (Refinement Object-Oriented Language) para la especificación formal.

16) Refactoring UML Models (Sunyé, 2001)

La evolución del software conlleva modificaciones que en ocasiones resulta complejo evaluar el impacto que tendrá en el comportamiento del sistema. La contribución del autor es emplear técnicas de refactorización para garantizar la preservación del comportamiento en los diagramas UML, con base en restricciones OCL. En el caso de la refactorización de diagramas de clases, se puede resumir en cinco operaciones: adición, eliminación, movimiento, generalización y especialización de elementos de modelado.

17) Towards Automating Source-Consistent UML Refactorings (Van Gorp et al., 2003)

El metamodelo UML no es suficiente para mantener la consistencia entre el diseño y el código, sobre todo cuando los sistemas informáticos evolucionan según las necesidades del usuario, al surgir estas necesidades, como consecuencia, provoca la generación de código maloliente. Por ello se propone como solución un conjunto de extensiones (método Pull Up y método Extract) al metamodelo UML llamada

GrammyUML, la cual permite con ayuda del lenguaje OCL mejorar el código oloroso, a la vez que se busca mantener la coherencia entre un modelo refactorizado y el código correspondiente. Como conclusión, la extensión probó ser eficiente, sin embargo, falta realizar pruebas en todos los escenarios posibles de refactorización, y por otro lado asegurar que la extensión propuesta sea la mejor opción para extender el UML.

18) On Proving the Correctness of Refactoring Class Diagrams of MDE Metamodels (Altoyan, 2023)

En este trabajo se trata el problema de demostrar la corrección de la refactorización de diagramas de clases en el ámbito de la Ingeniería Dirigida por Modelos (MDE), asegurando la parte semántica del metamodelo. En este trabajo los metamodelos se representan mediante los pares diagramas de clases y restricciones OCL, sin embargo, la metodología empleada se basa en usar Coq Proof Assistant, ignorando el lenguaje OCL, para probar formalmente que el proceso de refactorización de metamodelos MDE preserva la semántica original, sin embargo, los autores sugieren el uso de Álgebra Relacional para futuras especificaciones más efectivas.

III. Trabajos relacionados enfocados al uso de Modelos o Patrones de diseño.

1) A Design of an Assessment System for UML Class Diagram (Ali, 2007)

Se presenta UCDA (UML Class Diagram Assessor), una herramienta que evalúa diagramas de clases. Los diagramas evaluados fueron hechos por estudiantes con la herramienta de diseño Rational Rose. La evaluación consiste en asignar notaciones al diagrama del alumno (clases, objetos y relaciones). La lista de notaciones se compara con la lista de notaciones del diagrama de clases propuesto por el tutor, posteriormente los resultados se verifican comparando ambos diagramas de clases. El resultado es una lista de recomendaciones que guían a los estudiantes en la manera correcta de representar los diagramas de clases.

2) Automated Grading of Class Diagrams (Bian et al., 2019)

Se presenta un modelo para la calificación automatizada de diagramas de clases UML que busca coincidencias sintácticas, semánticas y estructurales, el cual consiste en mapear el diseño propuesto por el alumno y el diseño del instructor de la clase. El enfoque se basa en asignar pesos a cada elemento del diagrama de clases, como por ejemplo clases y atributos. Luego se comparan y se obtienen las puntuaciones, la calificación se obtiene por cada elemento que coincida perfectamente en ambos diseños. La herramienta propuesta TouchCORE, evalúa un caso de estudio real, y

arroja resultados con un 14% de diferencia entre la calificación asignada por el instructor. Además, con el modelo propuesto, se destaca la facilidad en el proceso de evaluación, ofreciendo al instructor la posibilidad de asignar calificaciones de manera automatizada.

3) *A Catalogue of Model Smells and Refactoring Operations for Object-Oriented Software (Sidhu et al., 2018)*

La ingeniería basada en modelos permite llevar a un mayor nivel de abstracción la evaluación de sistemas, lo que genera como resultado los olores de modelo, en este caso enfocados a diagramas de clases UML. Se consideran como olores los defectos de diseño que representan una mala calidad. Este estudio se basa en detectar esos malos olores en el diseño arquitectónico de sistemas, para ello se presenta un catálogo con los malos olores y la posible causa del problema, junto con algunas recomendaciones de operaciones de refactorización que los resuelven. Algunos de los olores que se mencionan son: Definitions of Classes with Equal Names, Ghost Class, Attribute Name Overridden, Equal Attributes in Sibling Classes, Large Class/ God Class, Long Parameter List, entre otros. Sin embargo, el proceso de refactorización puede introducir nuevos olores, lo que requiere realizar una evaluación intermedia entre los nuevos olores y la calidad del software. Se concluye que la falta de métodos de evaluación de diseños dificulta la evaluación en el proceso de refactorización.

4) *Evaluating the layout quality of UML class diagrams using machine learning (Bergström, 2022)*

En el proceso de desarrollo de software, los diseñadores recurren al uso de herramientas gráficas como lo son los diagramas UML, específicamente los diagramas de clases. Dado que estos diagramas reflejan la estructura interna del software, los autores consideran relevante garantizar la calidad en su diseño, ya que esta juega un papel muy importante en la comprensión del funcionamiento del software. Por ello los autores proponen un método automatizado que permite a través de imágenes evaluar la calidad en el diseño de diagramas de clases UML, haciendo uso de técnicas de Inteligencia Artificial como el Aprendizaje Automático. Este trabajo se considera por ir de la mano con este tema de investigación y por su aportación en identificar la alineación de los elementos de un diagrama de clases, así como los cruces y curvas entre las relaciones.

5) *Mistakes in UML Diagrams: Analysis of Student Projects in a Software Engineering Course (Chren, 2019)*

Se presenta un estudio realizado con diagramas UML con el propósito de identificar en un catálogo los defectos más comunes y evitar así la repetición de estos. Para ello se recopilieron los proyectos de estudiantes del área de Ingeniería de Software con el fin de identificar la frecuencia con que se comenten estos fallos. Al final, los investigadores ofrecen una lista como guía de apoyo de los defectos más observados, en el estudio se consideran diagramas de casos de uso, de actividades, de clases, máquinas de estado, de secuencia, de comunicación y de entidad relación, concluyendo que los diagramas de clases UML son los más complejos, seguido de los diagramas de máquinas de estado, y los diagramas de interacción. En las referencias se menciona un enlace para consultar el catálogo de errores, pero dado a que es una página restringida no se tiene acceso.

6) *Refactoring UML Models (Enckevort, 2009)*

UML desempeña un papel muy importante en el desarrollo de software, modificar un modelo UML antes de generar cualquier código es más barato que modificar tanto el modelo como el código fuente generado, sin embargo, no se cuentan con herramientas que permitan la refactorización de modelos UML. Por ello este trabajo propone un conjunto de métricas que ayude a detectar características no deseadas dentro del diagrama de clases UML, el cual se basa en la implementación de métricas de calidad orientadas a objetos, reglas de verificación sintácticas y semánticas del modelo descritas en lenguaje OCL. Entre las métricas propuestas constan de medir la profundidad de herencia, el número de hijos, el acoplamiento entre clases, la falta de cohesión en los métodos entre otras. También se incluye una lista con reglas sintácticas y semánticas. El trabajo concluye que el método es eficaz aplicado en este tipo de diagramas, el cual podría extenderse a otros modelos UML.

7) *Improving Model Driven Architecture with Requirements Models (Debnath, 2008)*

Este trabajo se basa en tomar las especificaciones de requerimientos en el ámbito empresarial y transformarlas en diagramas de clases a partir del lenguaje natural. Se usa CIMTool en el proceso para derivar un diagrama de clases a partir modelos de requisitos descritos en lenguaje natural. El enfoque propuesto facilita las primeras etapas del desarrollo de software en MDA al integrar modelos de requisitos en lenguaje natural. Los requerimientos de un modelo que describe un sistema empresarial son

capturados en lenguaje natural debido a que pueden ser interpretados por todas las partes involucradas, permitiendo la mejora de diagramas MDA.

8) *A tool-Supported Approach to Testing UML Design Models (Dinh-Trong et al., 2005)*

El enfoque propuesto en esta investigación se basa en el prototipo de la herramienta JAL, la cual permite comprobar modelos de diseño UML. Es sabido que el desarrollo de software impulsado por modelos es una herramienta útil en el área de ingeniería de software, pero para que estas herramientas tengan éxito, es necesario contar con técnicas de validación. Para ello este estudio se basa en probar diseños descritos mediante diagramas de clases UML para la parte estructural, y diagramas de actividades, para la parte dinámica. El proceso consiste en transformar el diseño UML en un formato ejecutable. Durante la fase de prueba se crean, modifican y observan el comportamiento del objeto, en la fase final se reportan las fallas.

9) *Model Smells In Uml Class Diagrams (Sidhu, 2016)*

Esta investigación se basa en identificar los olores de diseño de modelos en el proceso de desarrollo de software, partiendo de la idea de que la calidad del software depende de la calidad del modelo. Por otro lado, para asegurar la calidad del modelo se definen seis objetivos denominados 6C, como son: Correctness, Completeness, Consistency, Comprehensibility, Confinement y Changeability. También se listan varios ejemplos que presentan algún tipo de olor, entre los cuales se mencionan: Attribute Name Overridden, Concrete Superclass, Data Clumps, Inverted Operation Name, Large Class, Long Parameter List, Multiple Definitions of Classes with Equal Names, Primitive Obsession, Specialization Aggregation, Speculative Generality, Unnamed Element, Unused Element. El olor del modelo se identifica de manera manual, comparando y revisando diferentes métricas y patrones de diseño, lo que requiere tiempo, por ello se concluye que la implementación automática sería el siguiente paso.

10) *Impact of Design Patterns on Quantitative Assessment of Quality Parameters (Pradhan et al., 2015)*

El objetivo de este trabajo es proporcionar una metodología para medir el impacto de los atributos de calidad del software empleando patrones de diseño contra un sistema que no hace uso de ellos. Los patrones de diseño empleados son Bridge, Visitor y Abstract Factory. También se adopta el enfoque QMOOD (Quality Model for Object Oriented Design) para medir la calidad del software. La evaluación de los atributos de calidad del software se realiza antes de aplicar el patrón de diseño y después de

aplicarlo. Con este estudio se concluye que el uso de patrones de diseño mejora ciertos atributos de calidad, y que la metodología propuesta, proporciona una forma de cuantificar los atributos de calidad después de utilizar algún patrón de diseño.

IV. *Trabajos relacionados enfocados a técnicas de Aprendizaje Automático o Algoritmos de grafos.*

1) *Automatic Refactoring of Single and Multiple-view UML Models using Artificial Intelligence Algorithms (Baqais, 2016)*

El estudio consiste en refactorizar diagramas UML tanto de vista única como de vista múltiple a través de técnicas de IA (Artificial Intelligence). El enfoque de vistas múltiples se refiere a integrar diagramas de casos de uso, diagramas de secuencia y diagramas de clases. El objetivo es mejorar la calidad del diseño de software mediante operaciones de refactorización de acuerdo con métricas de calidad. Primero, se define el mejor algoritmo que se aplique en la refactorización de diagramas UML, se ajustan los parámetros para maximizar las métricas de calidad, por último, se evalúa y compara la calidad de los modelos UML refactorizados. Según el estudio, se observa una mejora en la calidad en modelos UML.

2) *Automatic classification of UML Class diagrams from images (Ho-Quang et al., 2014)*

El tema de estudio se basa en la clasificación y selección de diagramas a partir de 23 características, de las cuales 19 se consideran clave para la clasificación de diagramas de clases; también se mide la especificidad y sensibilidad de seis algoritmos de clasificación con un conjunto de 1300 imágenes. Las principales características para distinguir diagramas de clases de otras imágenes se basan en identificar rectángulos, líneas, la subdivisión de las clases, elipses, rombos, triángulos y la distribución de las formas. Una vez seleccionados los diagramas de clases se debe elegir el algoritmo de clasificación, posteriormente el entrenamiento del algoritmo y finalmente el análisis de los resultados. Se concluyó que el algoritmo Random Forest es el que mejor clasifica los diagramas de clases UML, mientras que con árboles de decisión obtuvieron la mayor puntuación en cuanto a especificidad. El trabajo de investigación plantea la mejora del rendimiento del clasificador basadas en el reconocimiento de texto.

3) *Automatic Generation of Sequence Diagram from Use Case Specification (Thakur, 2014)*

Se propone la herramienta AutoSDG (Auto Sequence Diagram Generator) para generar diagramas de secuencia a partir de especificaciones de casos de uso (UCS por sus siglas en inglés) escritas en lenguaje natural usando el analizador Stanford NL. El proceso consiste en leer la especificación de requisitos de software representado mediante diagramas de casos de uso, luego el API analizador de Stanford genera una estructura con etiquetas y dependencias, se aplican las Reglas de Estructura de Oraciones (SSR1-SSR15) y así generar un diagrama de secuencia. Se evalúan tres casos de estudio y se concluye en términos de exactitud y completitud que la herramienta genera mejores resultados a diferencia de otros enfoques, en este caso los resultados fueron comparados con los de la herramienta aToucan y solo se probó para oraciones simples en inglés.

4) *AnModeler: A Tool for Generating Domain Models from Textual Specifications (Thakur, 2016)*

Se presenta la herramienta AnMdeler, la cual permite generar modelos de análisis a partir de las especificaciones de los casos de uso. La herramienta utiliza el analizador Stanford para extraer dependencias de tipo y etiquetas de oraciones, utilizando un conjunto de reglas enmarcadas en base a los patrones verbales de Hornby. Finalmente, una vez identificadas las estructuras de las oraciones, la herramienta consolida los elementos de dominio en un diagrama de clases, de acuerdo con los resultados del estudio, esto se logra con un mayor grado de precisión y completitud en comparación con otras herramientas.

5) *Automatic builder of class diagram (ABCD): an application of UML generation from functional requirements (Abdessalem et al., 2015)*

Los modelos ayudan a comunicar mejor las ideas, partiendo de este enfoque, se propone la herramienta Automatic Builder of Class Diagram (ABCD), la cual consiste en tomar las especificaciones de requisitos del usuario en lenguaje natural (modelo origen), procesarlos (metamodelo fuente), y transformarlos al formato destino (archivo XMI). El proceso para generar diagramas de clases utiliza la herramienta de código abierto ArgoUML. Se concluye que ABCD produce mejores resultados al hacer la comparación con otras herramientas, sin embargo, la desventaja es que el sistema carece de mecanismos avanzados para tratar problemas con información redundante, es decir, una clase podría expresarse con sinónimos y en ese caso, el sistema no es

capaz de tratarlos como nombres de la misma clase, de modo que se generarían dos clases diferentes.

6) *Development of the Tool for Generation of UML Class Diagram from TwoHemisphere Model (Nikiforova, 2008)*

La idea de crear diagramas de clase parte de fusionar, por un lado, el proceso de negocio y por otro, el modelo conceptual, en el primer caso se toman los aspectos funcionales del sistema y por el otro, la estructura y relaciones de los datos. La lógica de dominio de aplicaciones o proceso de negocio se refiere a la manera en cómo definir el modelo fuente, para obtener un modelo objetivo, el cual estará definido como un diagrama de clases UML. Para ello se elige un enfoque de modelado de sistemas impulsado por modelos de dos hemisferios, Two-Hemisphere Model Driven (2HMD) con el objetivo de transformar los modelos de negocio y concepto en un diagrama de clases, aplicando reglas formales basadas en teorías de grafos.

7) *Decision tree-based Design Defects Detection (Maddeh et al., 2021)*

Los malos olores en un sistema son indudablemente un síntoma de una mala planificación, por ello se propone un enfoque para identificar defectos de diseño a un nivel de abstracción más alto, basado en diagramas UML, sin embargo, el problema es identificar primero el defecto, ya que cada diseñador puede interpretar ese error de manera personal. Por ejemplo, el defecto Feature Envy que hace referencia a métodos que pertenecen a otra clase en lugar de la clase fuente en la que fueron definidos, para un diseñador va a depender del número de llamadas a ese método. Sin embargo, los autores señalan que esta es una buena técnica de cómo puede identificarse un mal diseño. En este trabajo se consideran defectos de diseño y se crean árboles de decisión (ID3) para cada defecto. De acuerdo con los autores al usar este enfoque, se concluye que da un alto porcentaje en la detección de anomalías de diseño.

V. *Trabajos relacionados que se basan en el uso de métodos como son estudios empíricos o métricas de diseño orientado a objetos.*

1) *UML Class Diagrams - State of the Art in Layout Techniques (Eichelberger, 2003)*

El UML es un estándar ampliamente conocido para la especificación de software, a pesar de ello, las herramientas CASE no implementan los principios estéticos de diseño especialmente en diagramas de clases, sino que se basan en las reglas de dibujo de gráficos ignorando la parte semántica del diseño, lo que provoca resultados poco

legibles del diagrama. Para ello se propone un conjunto de principios estéticos para diagramas de clases como son: aplicar los criterios de jerarquías, eliminar dependencias implícitas, evitar el cruce de líneas y la superposición de elementos.

2) *On the Influence of UML Class Diagrams Refactoring on Code Debt: A Family of Replicated Empirical Studies (Freire et al., 2020)*

Detectar un modelo oloroso en la fase de diseño puede ayudar a evitar la propagación de errores en la fase de implementación. En consecuencia, esta investigación trata sobre la réplica del estudio empírico aplicado a procedimientos de refactorización de diagramas de clases UML. La implementación se basa en el uso de una herramienta de ingeniería inversa y una herramienta de identificación de problemas de código. El objetivo es determinar si los procedimientos de refactorización aplicados en los diagramas de clases UML mejora la calidad en el código que se genera automáticamente. El estudio permitió concluir que hay poca relación entre el modelo maloliente y el código.

3) *An Efficient Object Oriented Design Model: By Measuring and Prioritizing the Design Metrics of UML Class Diagram with Preeminent Quality Attributes (Rajesh, 2016)*

Este trabajo se enfoca en la mejora de la calidad del diseño del software mediante métricas de diseño de diagramas de clases UML desde las primeras fases del ciclo de vida del desarrollo de sistemas. Como solución se propone el modelo EOODM (Efficient Object Oriented Design Model), que ayuda a medir las métricas de calidad del diagrama de clases UML. Este modelo recopila las opiniones de los usuarios para identificar las necesidades del software. Luego se realiza un cálculo de las métricas de diseño orientado objetos enfocados al diagrama de clases, como cohesión, acoplamiento, composición; se asignan valores a las métricas en función del impacto en los factores de calidad del software como mantenibilidad, modificabilidad entre otros, y se da prioridad a aquellos atributos que obtuvieron mayor valor. Por último, se estima la modificabilidad del sistema. Se concluye que, con la ayuda de este modelo, es posible mejorar la calidad del diseño de software, reduciendo el impacto en modificaciones en la etapa de diseño.

4) *Similarity Assessment of UML Class Diagrams using Simulated Annealing (Al-Khiaty, 2014)*

Se trata sobre la duplicidad de artefactos y la complejidad generada para dar un mantenimiento adecuado teniendo modelos replicados, sin embargo, identificar

similitudes y diferencias en los modelos es el reto que se presenta, para dar solución a esto se plantea emplear el algoritmo “Simulated Annealing, con el fin de encontrar la correspondencia entre los elementos de dos modelos similares. Los aspectos de similitud son información léxica superficial como coincidencias en nombres (clases); información interna que mide las propiedades (atributos); similitud de comportamiento (métodos); e información de vecindad (relaciones). Las evaluaciones se realizan de manera empírica con base en tres métricas: precisión, recuperación y exactitud.

5) *Quality Improvement for UML and OCL Models through Bad Smell and Metrics Definition (Doan, 2019)*

Se presenta una extensión de la herramienta USE (UML-based Specification Environment). USE es una herramienta que evalúa modelos UML con el fin de mejorar la calidad del software en la etapa de diseño, mediante la detección de malos olores y la evaluación de calidad con base en métricas de diseño; se añade el análisis y la evaluación OCL (Object Constraint Language). La metodología de solución incluye una lista con 30 malos olores, cada olor se compara con el diseño evaluado, en caso de presentarse el olor se devuelve un valor verdadero junto con la lista de comentarios asociados a los elementos que causan el mal olor, también se permite que el usuario incorpore nuevos olores, según los requisitos del proyecto en particular (no es claro si de manera general un usuario puede agregar olores a la biblioteca o solo fue para un caso particular). El resultado de la evaluación le dirá al diseñador qué métricas se cumplen y cuáles no. USE se basa en el metamodelo de diseño de UML.

6) *A new approach for visualizing UML class diagrams (Gutwenger, 2003)*

Se propone una técnica para la mejora para la visualización de diagramas de clases UML jerárquicos y no jerárquicos de manera coherente y clara. Se menciona una lista de criterios estéticos para diagramas UML, como son minimizar el cruce de líneas, minimizar de curvas en las conexiones, utilizar una disposición ortogonal para asegurar que todas las flechas que indican relaciones dentro de un mismo componente sigan una misma dirección, evitar la anidación de jerarquías dentro de otras. Con este enfoque, se mejora la claridad de diagramas de clases UML y facilita la comprensión de la estructura del diagrama, combinando tanto elementos jerárquicos como no jerárquicos. Está técnica se plantea extender a otros tipos de diagramas UML.

7) *Methods, Algorithms and Software for the UML Class Diagram Refactoring (Deryugina, 2021)*

A mayor tamaño de un diagrama de clases, aumenta el tiempo de diseño y análisis, lo que hace necesario para el diseñador el uso de herramientas de software que proporcionen mecanismos automatizados que faciliten el proceso de diseño en las primeras fases del desarrollo de sistemas. Para ello se propone la herramienta UML Refactoring, el proceso de refactorización se lleva a cabo aplicando métricas de diseño orientado a objetos, como, por ejemplo, promedio de acoplamiento, profundidad de herencia, número de clases hijas, entre otras, y la aplicación de transformaciones usando los patrones de diseño fachada y estrategia. Además, se presenta un ejemplo de refactorización de diagramas de clases al cual se aplican transformaciones utilizando el patrón de diseño Fachada y la métrica complejidad estructural. Se concluye que la herramienta propuesta reduce significativamente la complejidad estructural del diagrama de clases.

8) *Case study: Class diagram restructuring. Transformation Tool Contest (Lano, 2013)*

Este caso de estudio trata sobre la refactorización de diagramas de clases UML con el objetivo de eliminar atributos duplicados. El proceso de transformación requiere identificar atributos repetidos entre clases, crear una subclase para esos atributos, y luego agruparlos en clases abstractas o clases nuevas que compartirán esas características, sin embargo, los problemas que se han identificado se basan en garantizar que la transformación produzca un único resultado, la finalización de la transformación, además de garantizar la correcta aplicación de las reglas. Este caso de estudio es útil como herramienta de prueba para evaluar la capacidad de transformación de diagramas de clases UML.

9) *An approach to refactoring of executable UML models (Dobrzanski, 2006)*

Un efecto negativo en la evolución del software es el deterioro de éste, mejor conocido como erosión del software. El artículo propone un enfoque sistemático para la refactorización de modelos UML ejecutables, la cual se propone como parte de la técnica de prevención de este proceso. Se denomina modelos ejecutables a los artefactos primarios en la metodología de desarrollo de software Agile MDA (Model-Driven Architecture), enfocándose tanto de la parte estructural como de comportamiento, es decir, en aplicar transformaciones que preserven el comportamiento, mejorando la calidad del diseño sin alterar su funcionalidad.

10) Guidelines on the aesthetic quality of UML class diagrams (Eichelberger, 2009)

El autor propone un experimento para determinar el impacto del uso de reglas de diseño de diagramas de clases UML en la comprensibilidad del usuario. El experimento sigue directrices propuestas según las reglas del UML y otras más basadas de acuerdo con la literatura, que sirven de guía para generar diagramas de clases UML, enfocándose en la calidad estética. Este trabajo se considera relevante porque trata aspectos relacionados con la estética de los diagramas de clases UML, sin embargo, el estudio solo se limita a analizar en cómo las pautas de diseño afectan el desempeño de las personas que trabajan con estos diagramas. La investigación resultó no concluyente debido a la baja cantidad de información, aunque a pesar de las dificultades se considera que los resultados reflejan que las reglas de diseño no tienen mayor impacto sobre el resto de las otras reglas. Lo que sugiere realizar el experimento con un número mayor de sujetos de prueba.

11) Ontology-Based Finite Satisfiability of UML Class Model (Hafeez, 2018)

Esta investigación trata la verificación del modelo de clases UML basada en ontologías, por un lado, se busca comprobar la consistencia y por otro, la satisfacibilidad de los modelos de clases. Este estudio propone un método de verificación para determinar la propiedad de correctitud en los diagramas de clases UML. Para procesar la ontología, el estudio se implementa en el marco de trabajo de Apache Jena.

12) On Evaluating the Layout of UML Class Diagrams for Program Comprehension (Dabo, 2005)

Este artículo clasifica los criterios considerados en el diseño de diagramas de clases UML, como son: minimizar el uso de cruces y curvas, dirección de la herencia, evitar la superposición, simetría, ortogonalidad, entre otros. Según los autores, en el contexto de la documentación de sistemas informáticos como representación gráfica de un diagrama UML, la disposición visual de los diagramas afecta su comprensión, pero esto depende de ciertos factores como son: la sintaxis y la semántica UML, la disposición espacial de los elementos dentro del diagrama y el conocimiento del dominio. El objetivo es proporcionar directrices de diseño a la hora de representar un diagrama, basados en los principios perceptuales para crear diagramas claros. Se presentan las ventajas y desventajas de aplicar estos criterios de diseño a dos herramientas de visualización de software y se dan algunas sugerencias para mejorar las herramientas.

12) Análise do uso da cor no Diagrama de Classes da Linguagem Unificada de Modelagem (UML) (Oliveira, 2020)

Un diagrama se representa como una serie de símbolos gráficos los cuales expresan cierto significado, utilizados como elementos verbales ya sea dentro del UML o de manera informal. En este artículo se aborda el uso de colores en el diagrama de clases UML, la técnica consiste en utilizar notas post-it con el fin de hacer más comprensible el diagrama. El uso de colores representa una herramienta de comunicación visual, capaz de organizar y jerarquizar datos, con el fin de comunicar mensajes, el cual cobra sentido una vez se le asigna cierto color. En el diseño gráfico, el color es importante para atraer o mantener la atención, así como para transmitir información o hacer que sea recordada. Estas anotaciones se consideran útiles para comunicar ideas y los colores ayudan a resaltar aspectos importantes, sin embargo, no se informa que cuáles colores o si existe un patrón a seguir. Es importante destacar que se recomienda el uso moderado ya que muchas personas no ven estos colores. Se observa que no hay una estandarización en el uso de las variables visuales, la aplicación del color, por ejemplo, queda a cargo de la subjetividad del "proyectista" o por la imposición de la herramienta utilizada en la modelación, Esto se evidencia también en la variación del grosor de las líneas y en el color de éstas, algunas son coloridas y otras no. El lenguaje verbal también presenta variaciones de tamaño y color. el color funciona como una extensión de la UML, son capas extras de información capaces de expresar más información sin ocupar más espacio. La falta de contraste y de grupos por color requiere una aproximación y más atención para identificar los diferentes tipos el uso de éstos. Para que el uso del color de forma aleatoria sea eliminado, será necesario que los profesionales del área adopten la práctica y hagan de esto un estándar universal.

13) Aesthetics of Class Diagrams (Eichelberger, 2003)

Este trabajo presenta una serie de características estéticas a considerar en el diseño de un diagrama de clases UML, tales como que los bordes deben ser dirigidos en una dirección común, centrar las clases padres con respecto a las clases hijo, evitar cruces y superposiciones en los bordes y restricciones generales sobre los bordes y los nodos, entre otros criterios que se listan. Ya que, de acuerdo con el autor, no hay un consenso sobre cómo producir diagramas legibles. Lo que hace necesario formular un catálogo con los criterios estéticos de diagramas de clases UML con el fin de estandarizar los diagramas. Con ello también se busca mejorar la comunicación entre usuarios con el fin de reducir malentendidos, y al mismo tiempo describir un conjunto de criterios estéticos que pueden conducir a una forma estandarizada de dibujar diagramas de clases UML.

II.3. Análisis de trabajos relacionados

De acuerdo con la revisión de los trabajos relacionados, la mayoría de los que fueron consultados se centran en la mejora de sistemas informáticos, con el objetivo de reducir la complejidad. Para ello, se proponen diferentes técnicas que ayuden en la refactorización a nivel de código, sin embargo, se deja a un lado la parte de diseño, restándole importancia al no abordar la mejora en cuestiones de legibilidad, siendo esta una parte importante a la hora de codificar, ya que un diseño complejo puede no ser comprensible para el programador (Sandoval, 2003). Para atender esta situación, se proponen diferentes enfoques, donde algunos autores sugieren nuevas métricas, otros usan lenguajes formales, otros más usan patrones de diseño, siendo estas algunas de las técnicas empleadas; todo ello con el fin de conseguir un software con el menor número de defectos. Bajo este escenario, la calidad visual de los materiales de aprendizaje, entre ellos los diagramas de clases UML, desempeñan un papel crucial en la educación. La búsqueda de la mejora estética de estos recursos como parte del desarrollo de software representa una parte muy importante, ya que con ello se busca garantizar la eficacia visual de los diagramas de clases UML, siendo este un tema que aún falta tratar más. Esto lleva a pensar que si existen estudios que buscan mejorar el software a nivel de código, también se requiere de métodos que permitan reestructurar diagramas UML, es decir, que ayuden a mejorar la calidad en cuestiones de diseño. De hecho, algunas características como el uso de colores, el grosor de líneas y el tamaño del texto se asocian con algunas deficiencias de diseño, y en cómo éstas afectan la parte visual de un diagrama de clases, sin embargo, no se sugieren métodos que ayuden en la reestructura de la legibilidad de diagramas de clases UML usados como recursos de aprendizaje. El término reestructura de la legibilidad de diagramas de clases UML, se refiere a mejorar aspectos asociados en la parte estética del diagrama sin modificar su arquitectura, lo que resulta conveniente para el usuario, tener diagramas que sean fáciles del leer. En la Tabla 1 se muestra un resumen comparativo de los trabajos relacionados; **Error! No se encuentra el origen de la referencia..**

Tabla 1. Trabajos relacionados con este tema de investigación.

Trabajo de investigación	Emplea Métricas de diseño	Emplea Modelo de calidad	Emplea Estándar UML	Es Servicio Web	Considera diseño
Automatic Generation of Sequence Diagram from Use Case Specification (Thakur, 2014)	No	No	Si	Si	No
AnModeler: A Tool for Generating Domain Models	No	No	Si	Si	No

Trabajo de investigación	Emplea Métricas de diseño	Emplea Modelo de calidad	Emplea Estándar UML	Es Servicio Web	Considera diseño
from Textual Specifications (Thakur, 2016)					
UML Class Diagrams - State of the Art in Layout Techniques (Eichelberger, 2003)	Si	No	Si	No	Si
Evolutionary layout of UML class diagrams (Gutenberg, 2006)	Si	No	Si	No	Si
Using Examples for Teaching Software Design: An Experiment Using a Repository of UML Class Diagrams (Karasneh, 2015)	No	No	Si	No	Si
On the Influence of UML Class Diagrams Refactoring on Code Debt: A Family of Replicated Empirical Studies (Freire et al., 2020)	No	No	Si	No	No
Impact of Design Patterns on Quantitative Assessment of Quality Parameters (Pradhan et al., 2015)	Si	Si	Si	No	No
UML diagram refinement (focusing on class- and use case diagrams) (Faitelson, 2017)	No	No	Si	No	Si
Automatic builder of class diagram (ABCD): an application of UML generation from functional requirements (Abdessalem et al., 2015)	No	No	Si	Si	No
Automatic Refactoring of Single and Multiple-view UML Models using Artificial Intelligence Algorithms (Baqais, 2016)	Si	No	Si	No	Si
A Design of an Assessment System for UML Class Diagram (Ali, 2007)	No	No	Si	Si	Si
Refactoring Class	No	No	Si	No	No

Trabajo de investigación	Emplea Métricas de diseño	Emplea Modelo de calidad	Emplea Estándar UML	Es Servicio Web	Considera diseño
Hierarchies with KABA (Streckenbach, 2004)					
Overview of Slicing and Feedback Techniques for Efficient Verification of UML/OCL Class Diagrams (Shaikh, 2018)	No	No	Si	No	Si
On the impact of layout Quality to Understanding UML Diagrams: Size Matters (Störrle, 2014)	No	No	Si	No	Si
A visual Base Framework for the Model Refactoring Techniques (Stolc, 2010)	No	No	Si	No	Si
A tool-Supported Approach to Testing UML Design Models (Dinh-Trong et al., 2005)	No	No	Si	No	No
Integrating and Refining UML Models (Liu et al., 2004)	Si	No	Si	No	No
Rule Modeling and Markup (Wagner, 2005)	No	No	No	No	No
An Efficient Object Oriented Design Model: By Measuring and Prioritizing the Design Metrics of UML Class Diagram with Preeminent Quality Attributes (Rajesh, 2016)	Si	No	Si	No	No
Automated Grading of Class Diagrams (Bian et al., 2019)	No	No	Si	Si	Si
Similarity Assessment of UML Class Diagrams using Simulated Annealing (Al-Khiaty, 2014)	Si	No	Si	No	No
Quality Improvement for UML and OCL Models through Bad Smell and Metrics Definition (Doan, 2019)	Si	No	Si	Si	No
Towards the Mechanized Semantics and Refinement	No	No	Si	No	No

Trabajo de investigación	Emplea Métricas de diseño	Emplea Modelo de calidad	Emplea Estándar UML	Es Servicio Web	Considera diseño
of UML Class Diagrams (Sheng, 2019)					
A Catalogue of Model Smells and Refactoring Operations for Object-Oriented Software (Sidhu et al., 2018)	Si	No	Si	No	Si
Development of the Tool for Generation of UML Class Diagram from TwoHemisphere Model (Nikiforova, 2008)	No	No	Si	No	No
A new approach for visualizing UML class diagrams (Gutwenger, 2003)	No	No	Si	No	No
UML Model Refactoring as Refinement: A Coalgebraic Perspective (Barbosa, 2008)	No	No	Si	No	Si
Decision tree-based Design Defects Detection (Maddeh et al., 2021)	Si	No	Si	No	Si
CUCKOO: Una plataforma web para la verificación de modelos UML (Caballé et al., 2009)	No	No	Si	Si	Si
Automatic classification of UML Class diagrams from images (Ho-Quang et al., 2014)	No	No	Si	Si	No
Methods, Algorithms and Software for the UML Class Diagram Refactoring (Deryugina, 2021)	Si	No	Si	Si	No
Refactoring OCL annotated UML class diagrams (Marković, 2007)	Si	No	Si	No	No
Case study: Class diagram restructuring. Transformation Tool Contest (Lano, 2013)	Si	No	Si	No	Si
Formal Refactoring for UML Class Diagrams. (Massoni, 2005)	No	No	Si	No	No
UML Model Refactoring with	No	No	Si	No	No

Trabajo de investigación	Emplea Métricas de diseño	Emplea Modelo de calidad	Emplea Estándar UML	Es Servicio Web	Considera diseño
Emphasis on Behavior Preservation (Hosseini, 2008)					
Practical Refactoring of Executable UML Models. (Dobrzanski, 2006).	No	No	Si	No	Si
Refactoring UML Models (Sunyé, 2001)	No	No	Si	No	Si
Towards Automating Source-Consistent UML Refactorings (Van Gorp, 2003)	Si	No	Si	No	Si
Evaluating the layout quality of UML class diagrams using machine learning (Bergström, 2022)	No	No	Si	No	Si
Guidelines on the aesthetic quality of UML class diagrams (Eichelberger, 2009)	No	No	Si	No	Si
On Proving the Correctness of Refactoring Class Diagrams of MDE Metamodels (Altoyan, 2023)	No	No	Si	No	Si
Mistakes in UML Diagrams: Analysis of Student Projects in a Software Engineering Course (Chren, 2019)	No	No	Si	No	Si
Refactoring UML Models (Enckevort, 2009)	Si	No	Si	Si	Si
Model Smells In Uml Class Diagrams (Sidhu, 2016)	Si	No	Si	No	Si
On Evaluating the Layout of UML Class Diagrams for Program Comprehension (Dabo, 2005)	No	No	Si	No	Si
Análise do uso da cor no Diagrama de Classes da Linguagem Unificada de Modelagem (UML) (Oliveira, 2020)	No	No	Si	No	Si
Aesthetics of Class Diagrams (Eichelberger,	No	No	Si	Si	Si

Trabajo de investigación	Emplea Métricas de diseño	Emplea Modelo de calidad	Emplea Estándar UML	Es Servicio Web	Considera diseño
2003)					
Reestructura de Recursos de Aprendizaje (Desiderio, 2025)	Si	Si	Si	Si	Si

En la Tabla 2 se muestra una breve descripción de los encabezados.

Tabla 2. Descripción de encabezados de la tabla de trabajos relacionados.

Encabezado	Descripción
Emplea métricas de diseño	Se usan métricas enfocadas al paradigma orientado a objetos o de diseño UML.
Emplea Modelo de calidad	Indica si se emplean marcos de referencia o estándares establecidos, que funcionan como puntos de referencia.
Emplea Estándar UML	Se aborda el uso de técnicas de modelado de objetos en ingeniería de software, usando herramientas gráficas o notaciones orientadas a objetos que permiten visualizar, construir y documentar un sistema.
Emplea Servicio Web	Se hace uso de aplicaciones de software que permiten la interacción interoperativa de máquina a máquina a través de una red.
Considera diseño	Se enfoca en la parte gráfica, útil para visualizar problemas de arquitectura.

Capítulo III

Metodología de solución

Este capítulo presenta la metodología de solución empleada en el proceso de reestructura de recursos de aprendizaje asociados con atributos de legibilidad.

III. Metodología de solución

En el capítulo anterior se mencionó que la mejora de recursos de aprendizaje es un área de investigación importante pero poco explorada, haciendo referencia a la investigación sobre la legibilidad de los diagramas de clases UML, centrándose en la calidad estética de estos, sin modificar su estructura interna, ya que con ello se busca mejorar aspectos relacionados con la comprensión. Entonces, la solución que se presenta para este trabajo consiste en crear un sistema que permite evaluar y reestructurar recursos de aprendizaje en formato PowerPoint denominado Automated Restructuring System, de aquí en adelante “ARE System”.

De manera general, el proceso de reestructuración inicia cuando el usuario carga al sistema un recurso de aprendizaje ya sea para evaluarlo o bien para reestructurarlo. Ambas acciones son independientes y se puede realizar una tarea sin necesidad de llevar a cabo la otra, lo que significa que se puede realizar la evaluación del recurso de aprendizaje sin necesidad de llevar a cabo el proceso de reestructura, o viceversa. Sin embargo, si previo a la reestructuración, se desea conocer particularmente los atributos con menor valor de correctitud, es necesario evaluar el recurso de aprendizaje con el propósito de identificar los atributos con menor ponderación, aunque esto no necesariamente implica que posteriormente se lleve a cabo el proceso de reestructura, ya que como se mencionó anteriormente, el sistema permite evaluar o reestructurar recursos de aprendizaje de manera independiente. En la parte de evaluación, las métricas aplicadas se rigen por el Modelo de Correctitud. Dicho modelo se describe en la sección III.1 y es el que define los atributos de calidad y parámetros que debe cumplir un diagrama de clases UML. Al evaluar un recurso de aprendizaje, los atributos que si se cumplen se marcan con una casilla de verificación, en caso contrario ésta aparecerá sin ninguna marca, el peso representa la importancia porcentual de cada atributo y el nombre corresponde a las características visuales tales como: el color de fondo y de línea, tamaño y color de fuente, ancho de línea y de las formas que representan las clases, así como de la claridad visual general de la diapositiva. La suma total de estos atributos ayuda a determinar la correctitud del recurso de aprendizaje. En la Ilustración 2 Ilustración 3 Ilustración 2 Ilustración 1 se muestra la lista de atributos que se evalúan en un recurso de aprendizaje.

Evaluation Results		
Weight	Criteria	Check
7%	Slide Background.	☒
14%	Line Color.	☒
10%	Line Width.	☐
8%	Shape width.	☐
7%	Shape color.	☐
10%	Font Size.	☒
10%	Font Color.	☐
34%	Not Blurring.	☒

Ilustración 2. Atributos de legibilidad reestructurados en un recurso de aprendizaje.

Para el proceso de reestructura, se presenta al usuario la lista de los atributos que se pueden cambiar, estos corresponden a características visuales tales como: colores de fondo y de línea, tamaño y color de fuente, ancho de las líneas y de las formas que representan las clases. Para ello el usuario deberá seleccionar de la lista los atributos que desea reestructurar, este proceso requiere al menos marcar un atributo de la lista para realizar las modificaciones pertinentes al recurso de aprendizaje, en caso contrario no se podrá llevar a cabo el proceso de reestructura. Las opciones que permiten reestructurar un recurso de aprendizaje se muestran en la Ilustración 3.

Process File

Select options required

- ☒ Slide Background
- ☒ Line Color
- ☒ Line Width
- ☒ Shape Background
- ☒ Text Size
- ☒ Text Color

Process Options

Ilustración 3. Porcentaje y Atributos de legibilidad evaluados en un recurso de aprendizaje.

En el diagrama de flujo de la Ilustración 4 se presenta de manera general la metodología de solución. Por otro lado, el proceso de reestructura, consiste en extraer los metadatos del recurso de aprendizaje, estos son leídos y manipulados por la biblioteca Apache POI, el resultado obtenido es un recurso de aprendizaje modificado y listo para descargarse.

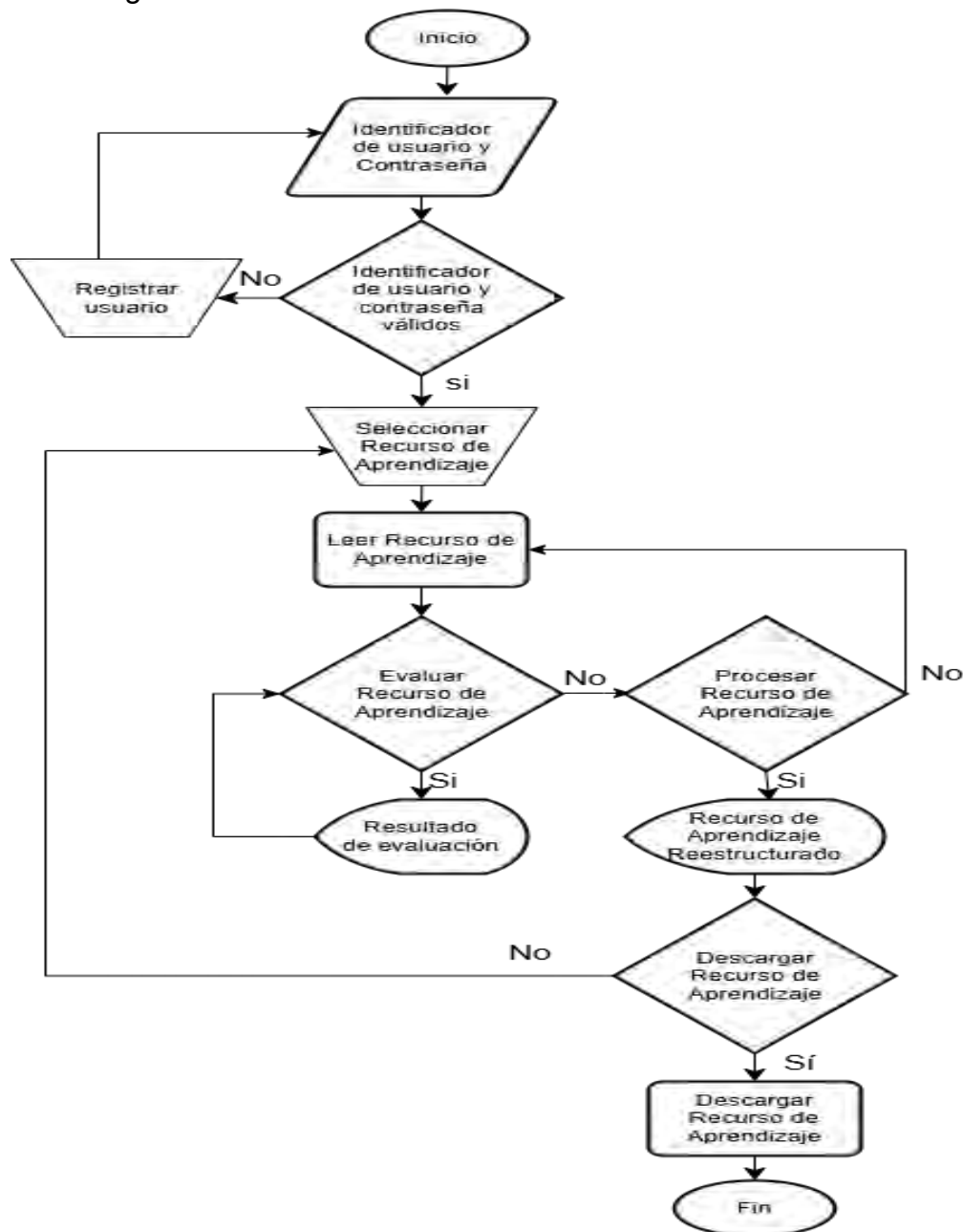


Ilustración 4. Metodología de solución propuesta

El desarrollo de la solución para la reestructura de diagramas de clases usados como recursos de aprendizaje consiste en los siguientes pasos:

1. Identificación del Modelo de Correctitud
2. Requerimientos del sistema
3. Análisis del sistema
 - a. Diagrama de Casos de uso
4. Diseño del sistema
 - a. Diagrama de Estados
 - b. Diagrama de Secuencia
 - c. Diagrama de Clases
 - d. Diagrama de Despliegue
5. Implementación
6. Pruebas
7. Resultados y hallazgos

III.1. Modelo de Correctitud

El sistema desarrollado realiza el proceso de reestructura de recursos de aprendizaje y se rige de acuerdo con los principios del Modelo de Correctitud (Sandoval, 2023). Dicho modelo sirve de guía para evaluar la calidad de los recursos de aprendizaje tomando en consideración el atributo principal denominado Correctitud. Este a su vez se clasifica en dos grupos, Correctitud en Dominio y Correctitud en Legibilidad, éstos a su vez se componen de otros atributos, los cuales se observan en la Ilustración 5. Con base en este modelo se determina el nivel de correctitud de la representación de un Diagrama de clases UML en presentaciones PowerPoint. Las partes de interés para el presente trabajo se destacan por un sombreado de color amarillo.

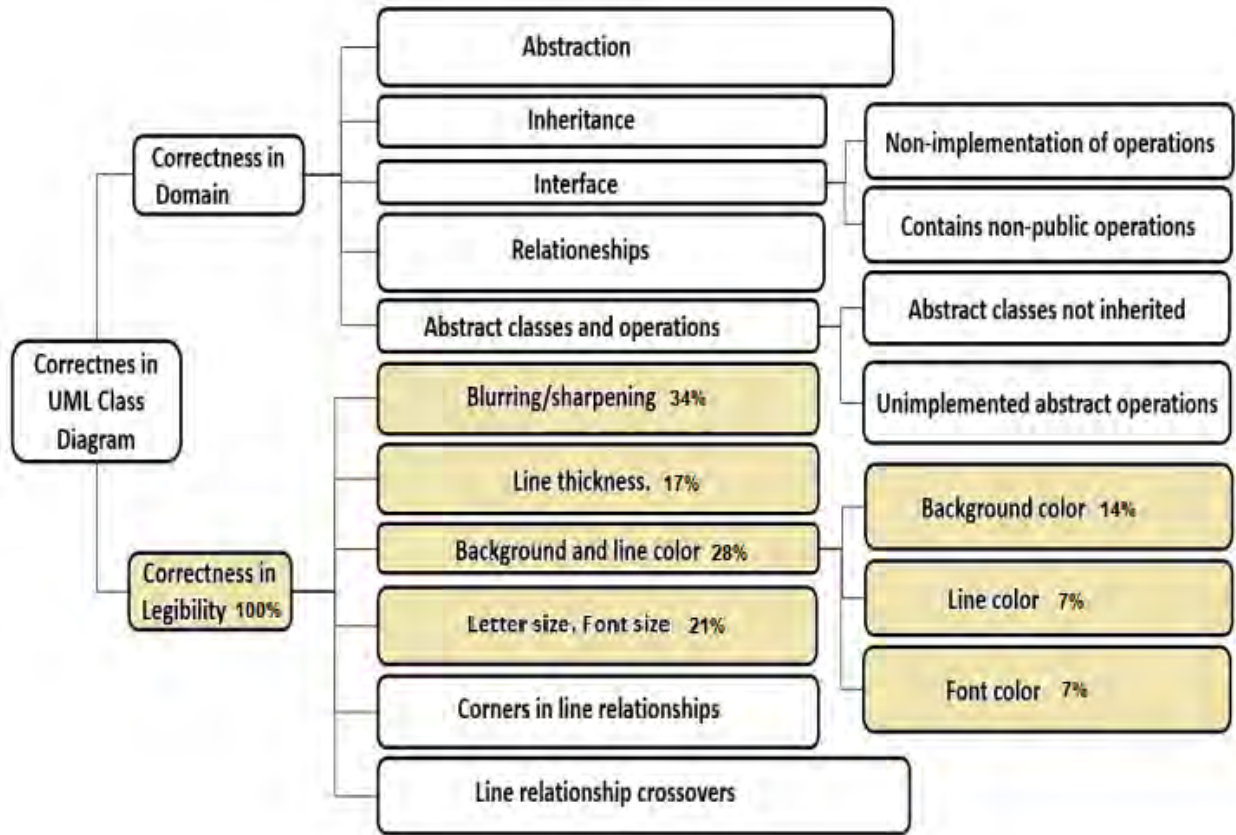


Ilustración 5. Modelo de Correctitud de diagramas de clases UML (Sandoval, 2023)

Los rubros que se tratan en este trabajo son los asociados al factor de Legibilidad, dentro de ellos, se encuentra el atributo nitidez, este atributo se considera debido a que un diagrama ilegible no puede ser comprensible al usuario. Por otro lado, también se destaca el grosor de los bordes de las clases, debido a que los bordes demasiado gruesos pueden oscurecer o dificultar la claridad visual del contenido de la clase (Störrle, 2014). Otros atributos se centran en la selección de colores, donde la principal recomendación es usar un fondo blanco con trazos negros para optimizar el contraste y facilitar la lectura (Rajngewerc, 2022). También se evalúa el tamaño de la fuente, sugiriendo un equilibrio en su proporción, evitar tamaños excesivamente pequeños o grandes y priorizar el uso de texto plano para mejorar la legibilidad (Garrido, 2017). Los subatributos antes mencionados son los que se consideraron en el desarrollo de esta herramienta, los cuales como se citó anteriormente, corresponden a los atributos de calidad que conforman el Modelo de Correctitud y constituyen la base de este trabajo de investigación, el cual atiende específicamente la reestructura de recursos de aprendizaje en el Dominio de Legibilidad.

En la Tabla 3 se presenta la correspondencia entre los parámetros de calidad que ARE System evalúa y los que pertenecen al Modelo de Correctitud, se distingue que no hay correspondencia uno a uno entre los atributos, esto es con la finalidad de aprovechar la

capacidad técnica de la herramienta y ofrecer al usuario una visión más específica de los elementos factibles a mejorar dentro del recurso de aprendizaje.

Tabla 3. Equivalencia de atributos

Atributo	
ARE System	Modelo de Correctitud
Slide Background	Background color
Shape background	
Line Color	Line color
Line width	Line thickness
Shape width	
Font size	Font size
Font color	Font color
Blurring	Blurring/Sharpening

Cada uno de los atributos que integran el Modelo de Correctitud tienen una ponderación que contribuye en la valoración de la correctitud del recurso de aprendizaje. Sin embargo, dado que en el presente trabajo no se consideran todos los atributos sino solo parte de ellos, las ponderaciones se adaptaron basándose en la relevancia de los valores propuestos en el Modelo de Correctitud, los cuales con la finalidad de lograr una valoración del 100% se obtuvieron aplicando una regla de tres. Estos valores una vez que se hayan incorporado todos los atributos al sistema de reestructura, se tendrá que hacer la actualización de los valores de correctitud con respecto al modelo original. Las ponderaciones utilizadas en este trabajo se pueden consultar en la Tabla 4.

Tabla 4. Ponderación de los atributos de calidad que evalúa ARE System

Atributo	% Ponderación
Slide Background	7
Shape background	7
Line Color	14
Line width	10
Shape width	8
Font size	10
Font color	10
Blurring	34
Total	100

Finalmente, el porcentaje total obtenido se interpreta como el nivel de Correctitud del recurso de aprendizaje, en cuanto al factor de Legibilidad del Modelo de Correctitud, donde una puntuación de 100% indica un recurso de aprendizaje que cumple con todos los atributos de calidad.

III.2. Requerimientos del sistema

Como fundamento base para el desarrollo del software, se establecieron los requerimientos del sistema tomando en cuenta los alcances y necesidades para lograr el objetivo del sistema de reestructura. En la Tabla 5 se presentan los requerimientos funcionales que el sistema ARE System debe cumplir.

Tabla 5. Requerimientos Funcionales

No.	ID	Descripción
1	RF001	El sistema al iniciar debe mostrar una breve descripción de éste.
2	RF002	El sistema debe autenticar a un usuario mediante sus credenciales de acceso (identificador de usuario, contraseña).
3	RF003	El sistema debe mostrar al usuario un mensaje de bienvenida cuando el ingreso se ha logrado exitosamente.
4	RF004	El sistema debe mostrar al usuario una notificación indicando si el ingreso ha sido exitoso o no.
5	RF005	El sistema debe redirigir a la página de <i>Inicio</i> cuando el proceso de <i>Inicio de Sesión</i> no sea exitoso.
6	RF006	El sistema debe permitir a nuevos usuarios registrarse capturando los siguientes datos: CURP, Nombre, Apellido y Clave.
7	RF007	El sistema debe almacenar en la base de datos la información del nuevo usuario.
8	RF008	El sistema debe enviar una notificación indicando si el alta del usuario ha sido exitosa o no.
9	RF009	El sistema debe asignar automáticamente un identificador de usuario a nuevos usuarios.
10	RF010	El sistema debe permitir al usuario cargar el archivo que contiene el recurso de aprendizaje.
11	RF011	El sistema debe alojar el archivo que contiene el recurso de aprendizaje en formato PowerPoint.
12	RF012	El sistema debe mostrar la lista de archivos que se han cargado al sistema.
13	RF013	El sistema debe mostrar una notificación indicando que el archivo no se ha cargado exitosamente.
14	RF014	El sistema debe evaluar los atributos de calidad asociados al factor de Legibilidad del recurso de aprendizaje.
15	RF015	El sistema debe presentar los porcentajes de evaluación de los atributos de calidad asociados al factor de Legibilidad.
16	RF016	El sistema debe permitir a los usuarios seleccionar los atributos para aplicar los cambios asociados al recurso de aprendizaje.
17	RF017	El sistema debe reestructurar el recurso de aprendizaje de acuerdo con los criterios de calidad seleccionados por el usuario.
18	RF018	El sistema debe generar un recurso de aprendizaje en formato

No.	ID	Descripción
		PowerPoint de acuerdo con los criterios de calidad seleccionados por el usuario.
19	RF019	El sistema debe mostrar una previsualización del recurso de aprendizaje con los cambios aplicados.
20	RF020	El sistema debe permitir al usuario descargar el recurso de aprendizaje con los cambios aplicados en formato PowerPoint.

III.3. Análisis del sistema

La siguiente sección presenta el proceso de desarrollo del sistema ARE System de acuerdo a los requisitos previamente establecidos y cómo se implementaron.

1) *Diagrama de casos de uso*

En la Ilustración 6 se presenta el diagrama de casos de uso del sistema, este comienza cuando un usuario ingresa en la aplicación. Para poder ingresar al sistema es necesario que el usuario haya realizado su registro previamente, para ello se requiere del caso de uso *Registrar Usuario*, el cual permite dar de alta a nuevos usuarios, una vez completado este proceso, el caso de uso *Iniciar Sesión* le dará acceso al sistema, esto bastará con ingresar sus datos. Para el tratamiento de los recursos de aprendizaje se describe el caso de uso *Cargar Archivo*, el cual permite cargar el recurso de aprendizaje al sistema. De acuerdo con las reglas del Modelo de Correctitud, el caso de uso *Evaluar Recurso* realiza una estimación sumando aquellos atributos que se cumplen, y de acuerdo con los resultados obtenidos se determina el nivel de correctitud del recurso de aprendizaje. Por otro lado, *Procesar Recurso*, permite realizar las modificaciones al recurso de aprendizaje de acuerdo con los criterios de calidad seleccionados por el usuario. De éste se incluye el caso de uso *Generar Recurso*, el cual proporciona el recurso de aprendizaje con los cambios aplicados. A su vez, se extiende del caso de uso *Descargar Archivo*, dejando abierta la opción al usuario para disponer del recurso de aprendizaje.

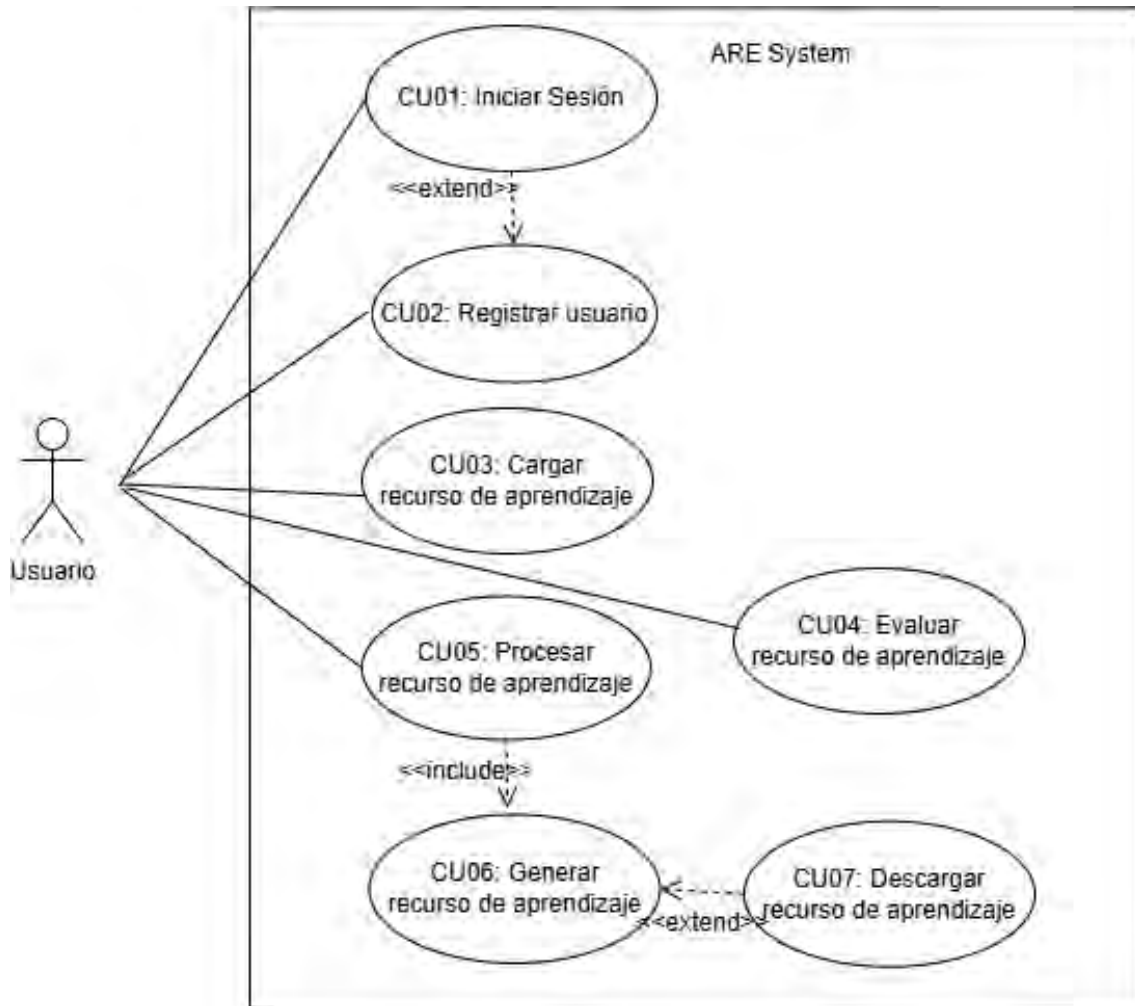


Ilustración 6. Diagrama de Casos de uso ARE System

En resumen, este sistema está diseñado para gestionar la interacción del usuario con elementos como son recursos de aprendizaje en archivos de PowerPoint, ofreciendo funcionalidades como cargar, evaluar, procesar y exportar dichos recursos. Para más información sobre el uso del sistema, consulte los anexos al final de este documento. Refiérase al Anexo A. Manual de usuario.

2) Descripción de los casos de uso

De la Tabla 6 a la Tabla 12, se describe el proceso de implementación de las actividades que los usuarios pueden realizar, los cuales están asociados a cada Caso de uso descrito en la sección anterior.

Tabla 6. Descripción del Caso de Uso Iniciar Sesión

1	ID	CU001
2	Nombre del Caso de uso	Iniciar Sesión

3	Autor	Areliá Desiderio Patiño	Última modificación	1.7
4	Fecha de creación	20/05/2024	Fecha de última modificación	04/03/2025
5	Actores	Usuario		
6	Descripción	Este caso de uso describe los pasos necesarios para el proceso de autenticación mediante las credenciales de inicio de sesión (identificador de usuario y contraseña).		
7	Precondición	El usuario debe estar dado de alta en el sistema.		
8	Postcondiciones	El sistema muestra la página menú de operaciones.		
9	Escenario principal de éxito	1. El sistema muestra el formulario de inicio de sesión. 2. El usuario ingresa sus credenciales de acceso: identificador y contraseña. 3. El sistema comprueba los datos del usuario en la base de datos. 4. El sistema muestra un mensaje de bienvenida.		
10	Escenario alternativo (1)	1. El sistema muestra el formulario de inicio de sesión. 2. El usuario ingresa sus credenciales de acceso: identificador y contraseña. 3. El sistema comprueba los datos en la base de datos. 4. Se muestra un mensaje indicando que las credenciales de inicio de sesión no son válidas. 5. El usuario ingresa nuevamente sus credenciales de acceso: identificador y contraseña. 6. El sistema muestra un mensaje de bienvenida.		
11	Escenario de fracaso	1. El sistema muestra el formulario de inicio de sesión. 2. El usuario ingresa sus credenciales de acceso: identificador y contraseña. 3. El sistema comprueba los datos en la base de datos. 4. El sistema muestra un mensaje informando que los datos no son válidos.		
12	Casos de uso extendido	CU01. Registrar usuario		
13	Casos de uso incluido	CU08. Cerrar sesión		
14	Comentarios	Ninguno		

Tabla 7. Descripción del Caso de Uso Registrar Usuario

1	ID	CU02		
2	Nombre del Caso de Uso	Registrar Usuario		
3	Autor	Areliá Desiderio Patiño	Última Modificación	1.7

4	Fecha de Creación	20/05/2024	Fecha de última modificación	04/03/2025
5	Actores	Usuario		
6	Descripción	Este caso de uso describe la implementación necesaria para registrar un nuevo usuario al sistema.		
7	Precondición	El sistema debe estar desplegado en el siguiente enlace http://localhost:8080/are/frontPage.jsp# .		
8	Postcondiciones	El alta del usuario se ha realizado con éxito en el sistema.		
9	Escenario principal de éxito	1. El sistema muestra el formulario de alta. 2. El usuario captura los siguientes datos: a. Nombre b. Apellidos c. CURP d. Contraseña e. Confirmación de contraseña 3. El sistema valida la información. 4. La información capturada se guarda en la base de datos. 5. El sistema envía un mensaje confirmando el alta al usuario.		
10	Escenario alternativo (1)	1. El sistema muestra el formulario de alta. 2. El usuario captura los siguientes datos: a. Nombre b. Apellidos c. CURP d. Contraseña e. Confirmación de contraseña 3. Se verifica la información en la base de datos. 4. Se muestra una notificación indicando que los datos ya están en uso. 5. El usuario captura los nuevos datos. 6. El sistema envía un mensaje confirmando el alta del usuario.		
11	Escenario de fracaso	1. El sistema muestra el formulario de registro. 2. El usuario captura los siguientes datos: a. Nombre b. Apellidos c. CURP d. Contraseña e. Confirmación de contraseña 3. El sistema valida la información. 4. La información capturada se verifica en la base de datos. 5. El sistema envía un mensaje indicando que ya existen los datos capturados.		
12	Caso de uso extendido	CU02. Iniciar sesión		

13	Caso de uso incluido	Ninguno
14	Comentarios	Ninguno

Tabla 8. Descripción del Caso de Uso Cargar Archivo

1	ID	CU03		
2	Nombre del Caso de Uso	Cargar Archivo		
3	Autor	Arelia Desiderio Patiño	Última Modificación	1.2
4	Fecha de Creación	20/05/2024	Fecha de última modificación	14/01/2025
5	Actores	Usuario		
6	Descripción	Este caso de uso describe los pasos necesarios para que un usuario cargue al sistema un archivo que contiene el recurso de aprendizaje.		
7	Precondición	El sistema debe estar disponible para recibir el archivo que contiene el recurso de aprendizaje.		
8	Postcondiciones	El sistema queda preparado con el recurso de aprendizaje.		
9	Escenario principal de éxito	1. El sistema muestra el cuadro de diálogo para seleccionar el archivo. 2. El usuario selecciona el archivo que contiene el recurso de aprendizaje. 3. El archivo se almacena en el sistema. 4. El sistema muestra una notificación de recibido.		
10	Escenario de fracaso	1. El sistema muestra el cuadro de diálogo para seleccionar el archivo. 2. El usuario selecciona el archivo que contiene el recurso de aprendizaje. 3. El sistema valida el archivo. 4. El sistema no guarda el archivo en el servidor. 5. El sistema muestra un mensaje informando que ocurrió un error.		
11	Caso de uso extendido	Ninguno		
12	Caso de uso incluido	Ninguno		
13	Comentarios	Ninguno		

Tabla 9. Descripción del Caso de Uso Evaluar Recurso

1	ID	CU04
2	Nombre del Caso de Uso	Evaluar recurso

3	Autor	Arelia Desiderio Patiño	Última Modificación	1.2
4	Fecha de Creación	20/05/2024	Fecha de última modificación	14/01/2025
5	Actores	Usuario		
6	Descripción	Este caso de uso describe los pasos necesarios para la evaluación de los atributos de calidad del recurso de aprendizaje.		
7	Precondición	El recurso de aprendizaje debe estar disponible en el sistema.		
8	Postcondiciones	El sistema muestra los resultados de la evaluación del recurso de aprendizaje.		
9	Escenario principal de éxito	1. El sistema lee el contenido del archivo. 2. El sistema analiza el contenido del archivo. 3. El sistema muestra el resultado de evaluación del recurso de aprendizaje.		
10	Escenario alternativo (1)	1. El sistema no puede leer el contenido del archivo. 2. El sistema muestra un mensaje indicando que hay problemas para leer el archivo. 3. El usuario selecciona un nuevo archivo y reenvía el archivo que contiene el recurso de aprendizaje.		
11	Escenario de fracaso	1. El sistema no puede leer el recurso de aprendizaje, no existe o está dañado. 2. El sistema muestra una notificación informando que ocurrió un problema al leer el archivo.		
12	Caso de uso extendido	Ninguno		
13	Caso de uso incluido	Ninguno		
14	Comentarios	Ninguno		

Tabla 10. Descripción del Caso de Uso Procesar Recurso

1	ID	CU05		
2	Nombre del Caso de Uso	Procesar recurso		
3	Autor	Arelia Desiderio Patiño	Última Modificación	1.2
4	Fecha de Creación	20/05/2024	Fecha de última modificación	14/01/2025
5	Actores	Usuario		
6	Descripción	Este caso de uso describe los pasos necesarios para reestructurar el archivo que contiene el recurso de		

		aprendizaje.
7	Precondición	La lista de los atributos de calidad seleccionados por el usuario. Estos pueden ser el color de fondo de la diapositiva, colores en las líneas, ancho de las líneas, color de fondo de las clases, tamaño de la fuente o color de fuente, los cuáles determinan el proceso de reestructura.
8	Postcondiciones	El sistema muestra una vista previa del recurso de aprendizaje con los cambios aplicados, los atributos se reestructuran de acuerdo con la lista de atributos seleccionados por el usuario.
9	Escenario principal de éxito	1. El sistema lee la lista de atributos de calidad que definirá el proceso de reestructura del recurso de aprendizaje. 2. El sistema aplica los cambios necesarios al recurso de aprendizaje según las opciones del usuario. 3. El sistema muestra un mensaje informando que el proceso de reestructura ha sido exitoso.
10	Escenario de fracaso	1. El sistema no recibe la lista de atributos de calidad que definirá el proceso de reestructura del recurso de aprendizaje. 2. El sistema no modifica el contenido del archivo. 3. El recurso de aprendizaje con los cambios no se genera. 4. El sistema muestra un mensaje informando que el proceso de reestructura no se puede realizar.
11	Caso de uso extendido	CU06. Generar recurso
12	Caso de uso incluido	Ninguno
13	Comentarios	Ninguno

Tabla 11. Descripción del Caso de Uso Generar Recurso

1	ID		CU06	
2	Nombre del Caso de Uso		Generar recurso	
3	Autor	Arelia Desiderio Patiño	Última Modificación	1.2
4	Fecha de Creación	20/05/2024	Fecha de última modificación	14/01/2025
5	Actores	Usuario		
6	Descripción	Este caso de uso describe los pasos necesarios para generar un archivo con el recurso de aprendizaje reestructurado.		
7	Precondición	El sistema debe disponer del recurso de aprendizaje.		

8	Postcondiciones	Generar el recurso de aprendizaje con los cambios aplicados.
9	Escenario principal de éxito	1. El sistema analiza el contenido del archivo. 2. El sistema genera una copia del recurso de aprendizaje procesado. 3. El sistema muestra una notificación indicando que el recurso de aprendizaje reestructurado ha sido creado.
10	Escenario de fracaso	1. El sistema analiza el contenido del archivo. 2. El sistema muestra un mensaje informando que ocurrió un error durante el procesamiento.
11	Caso de uso extendido	CU07. Descargar archivo
12	Caso de uso incluido	Ninguno
13	Comentarios	Ninguno

Tabla 12. Descripción del Caso de Uso Descargar Archivo

1	ID		CU07	
2	Nombre del Caso de Uso		Descargar Archivo	
3	Autor	Arelia Desiderio Patiño	Última Modificación	1.2
4	Fecha de Creación	20/05/2024	Fecha de última modificación	14/01/2025
5	Actores	Usuario		
6	Descripción	Este caso de uso describe los pasos necesarios para exportar un archivo a la computadora del usuario.		
7	Precondición	El sistema debe disponer del recurso de aprendizaje modificado.		
8	Postcondiciones	El sistema muestra el menú de operaciones para realizar la siguiente petición.		
9	Escenario principal de éxito	1. El usuario solicita descargar el recurso de aprendizaje procesado. 2. El sistema muestra un mensaje indicando que el recurso de aprendizaje reestructurado ha sido descargado.		
10	Escenario de fracaso	1. El usuario solicita la descarga del archivo modificado. 2. El sistema encuentra un error al intentar leer el archivo. 3. El sistema muestra un mensaje de error indicando que hubo un problema durante el procesamiento.		
11	Caso de uso extendido	Ninguno		
12	Caso de uso incluido	Ninguno		

13	Comentarios	Ninguno
----	-------------	---------

III.4. Diseño del sistema

1) **Descripción general del diagrama de estados.**

El esquema se muestra en la Ilustración 7.

1. Mostrando interfaz de acceso.
Se espera que el usuario ingrese sus credenciales de acceso.
2. Comprobando usuario y contraseña.
Si los datos son correctos el flujo continúa al estado de solicitar recurso.
3. Guardando información en la base de datos.
Si los datos no existen en la base de datos, el usuario deberá registrarse.
4. Solicitando recurso de aprendizaje.
Se espera que el usuario cargue el archivo PowerPoint con el recurso de aprendizaje al sistema.
5. Verificando recurso de aprendizaje.
Se valida el archivo PowerPoint con el recurso de aprendizaje.
6. Leyendo recurso de aprendizaje.
Se analiza el contenido del archivo PowerPoint.
7. Decisión de flujo de acción.
El usuario tendrá que elegir entre evaluar o procesar el archivo.
8. Procesando recurso de aprendizaje.
Se modifica el contenido del recurso de aprendizaje.
9. Evaluando recurso de aprendizaje.
Se analiza el contenido del recurso de aprendizaje.
10. Mostrando resultados de evaluación.
Se muestran en pantalla los resultados correspondientes a la evaluación.
11. Descargando recurso de aprendizaje.
El recurso de aprendizaje está listo para su descarga.

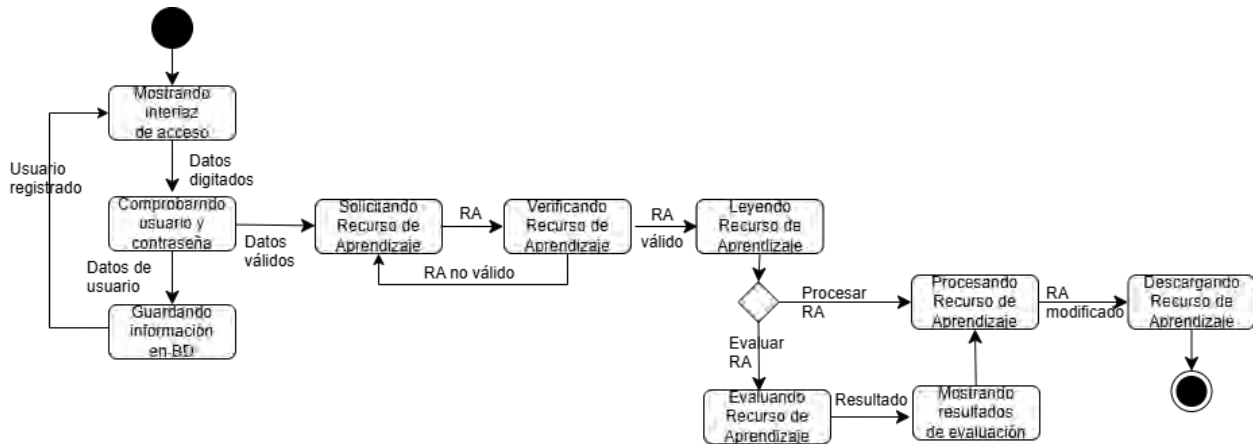


Ilustración 7. Diagrama de Estados de ARE System

2) Descripción general del diagrama de secuencia.

Los esquemas se muestran de la Ilustración 8 a la Ilustración 12.

1. Interacción inicial

- Ingreso al sistema: El usuario ingresa al sistema.
- Muestra opciones ingresar o registrar: El sistema presenta al usuario las opciones de registro (alta de un nuevo usuario) y login (inicio de sesión para usuarios ya registrados).
- El usuario escoge una opción: El usuario selecciona una de las opciones presentadas.

2. Opciones alternas

- Registro de un nuevo usuario
 - Se presenta formulario de registro: El usuario captura la información solicitada.
 - Validar información: El sistema valida la información proporcionada.
 - Registro exitoso: La base de datos confirma que el registro fue exitoso.
 - Mensaje de notificación: Se muestra una notificación indicando el resultado de la operación.
- Inicio de sesión
 - El sistema muestra el formulario: El usuario captura la información solicitada.
 - Validar datos (usuario, contraseña): El sistema valida la información.
 - Registro exitoso: El sistema permite el ingreso al usuario.
 - Menú de operaciones: El sistema presenta las opciones del sistema.

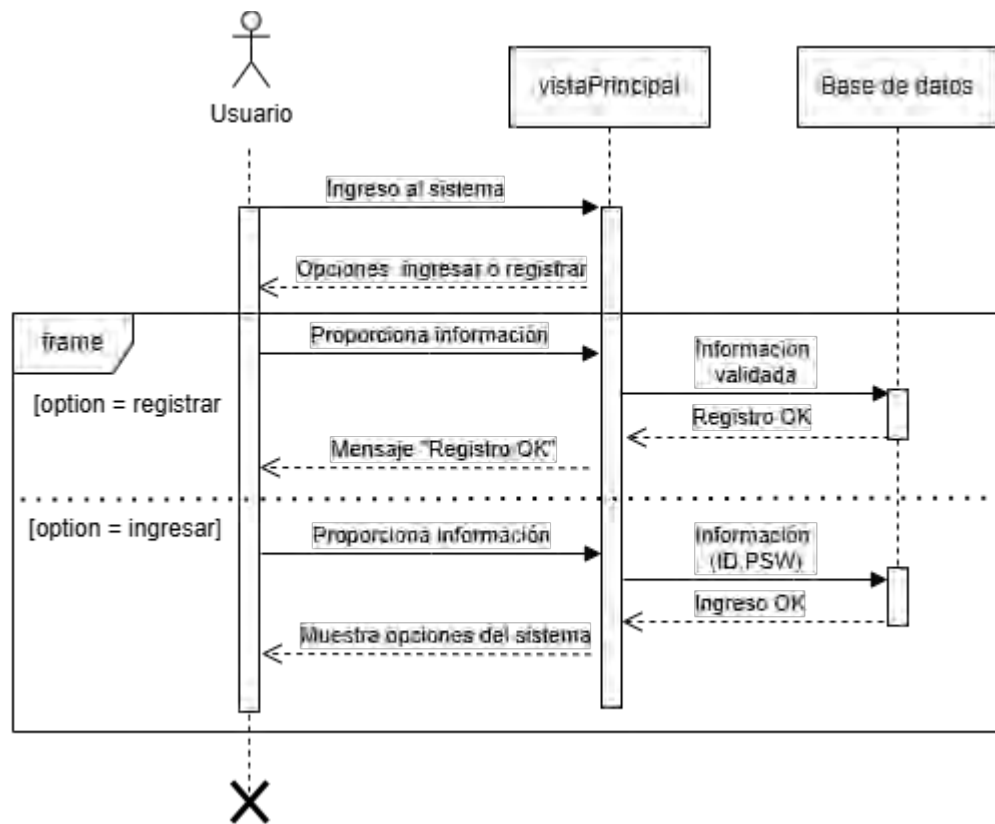


Ilustración 8. Diagrama de Secuencia de ARE System

3. Opciones alternas (Toda vez que el usuario ha ingresado al sistema).

- Subir archivo
 - Upload File: El usuario importa un archivo al sistema.
 - Archivo recibido: El sistema notifica el resultado de la operación.

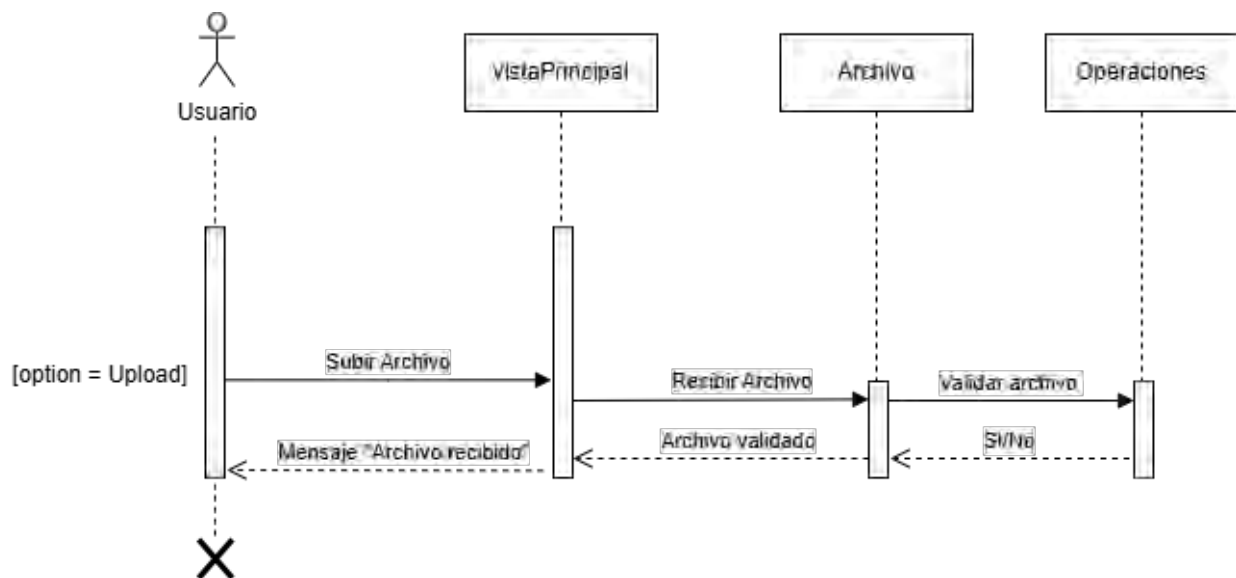


Ilustración 9. Diagrama de Secuencia de ARE System (Opción Upload)

- Evaluar archivo
 - Evaluar archivo: Se realiza un análisis de los atributos de calidad del recurso de aprendizaje.

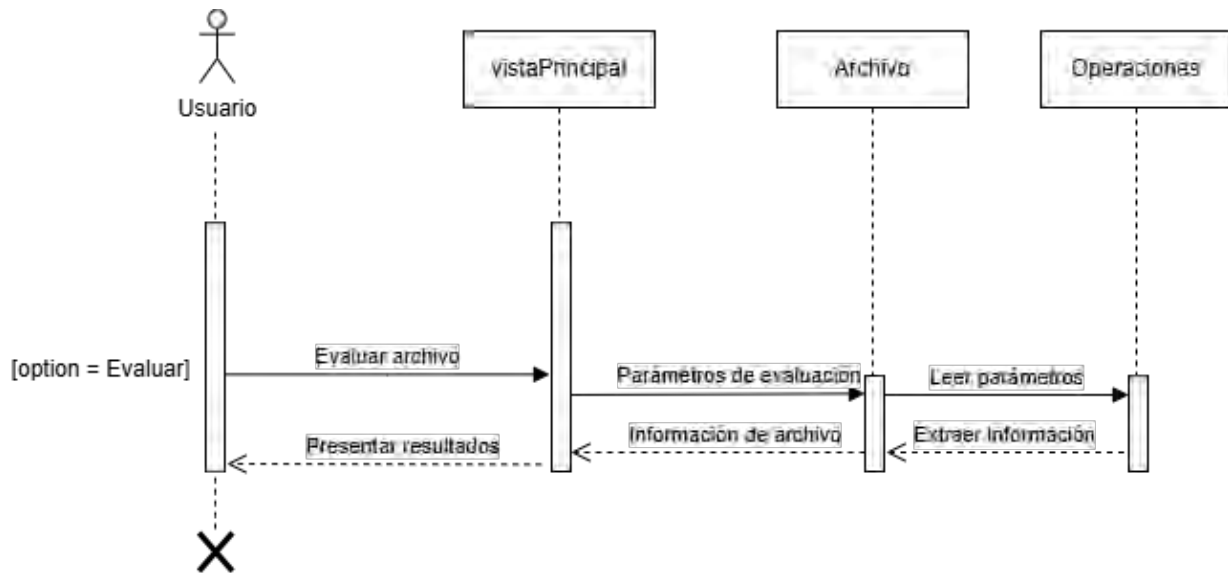


Ilustración 10. Diagrama de Secuencia de ARE System (Opción Evaluar)

- Procesar archivo
 - Reestructurar archivo: El sistema modifica el recurso de aprendizaje de acuerdo con los atributos de calidad seleccionados.

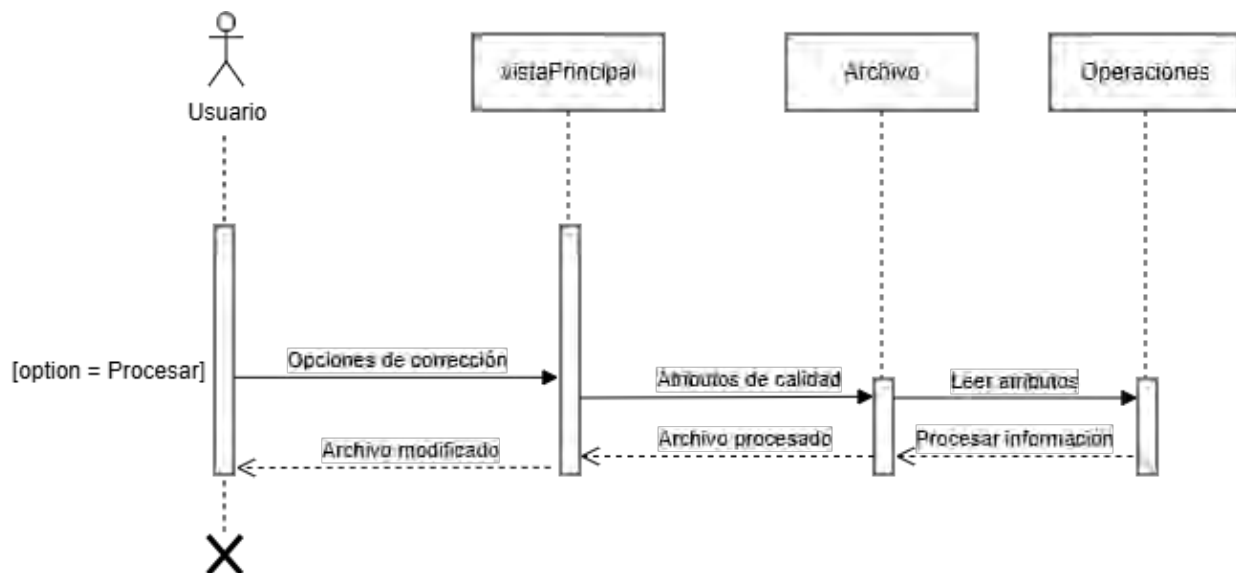


Ilustración 11. Diagrama de Secuencia de ARE System (Opción Procesar)

- Descargar archivo
 - Download file: El archivo se almacena en el dispositivo del usuario.

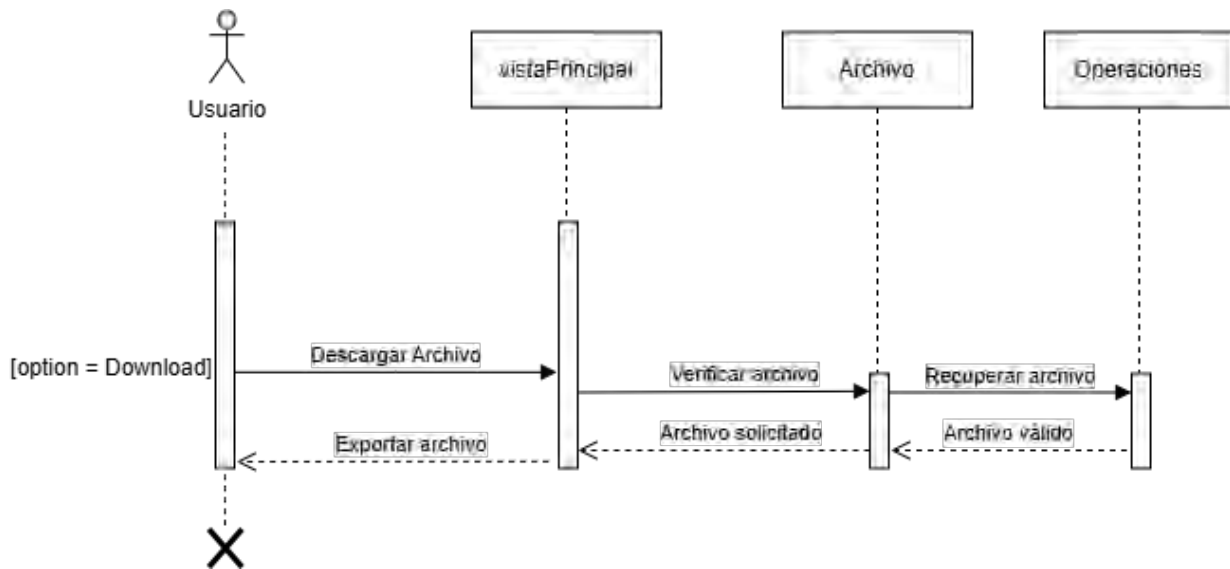


Ilustración 12. Diagrama de Secuencia de ARE System (Opción Download)

3) Descripción general del diagrama de clases.

El esquema se muestra en Ilustración 13.

1. Archivo
 - Clase que se encarga de la gestión de archivos dentro del sistema. Se conecta con LR_actual y LR_modificado, indicando la gestión de esos archivos en su estado original y después de ser procesados.
2. Operaciones
 - Clase responsable de realizar los procesos principales del sistema.
3. Usuario
 - Clase que representa a los usuarios que interactúan con el sistema.
4. LR_actual y LR_modificado
 - Clases que se encargan del recurso de aprendizaje, antes de ser procesado (actual) y después de ser procesados (modificado).
5. Vistas
 - vistaPrincipal: Clase que sirve como interfaz del sistema.
 - vistaGestorLR: Clase que gestiona las correcciones del sistema.
 - vistaGestorUsuario: Clase que gestiona la administración de los usuarios.

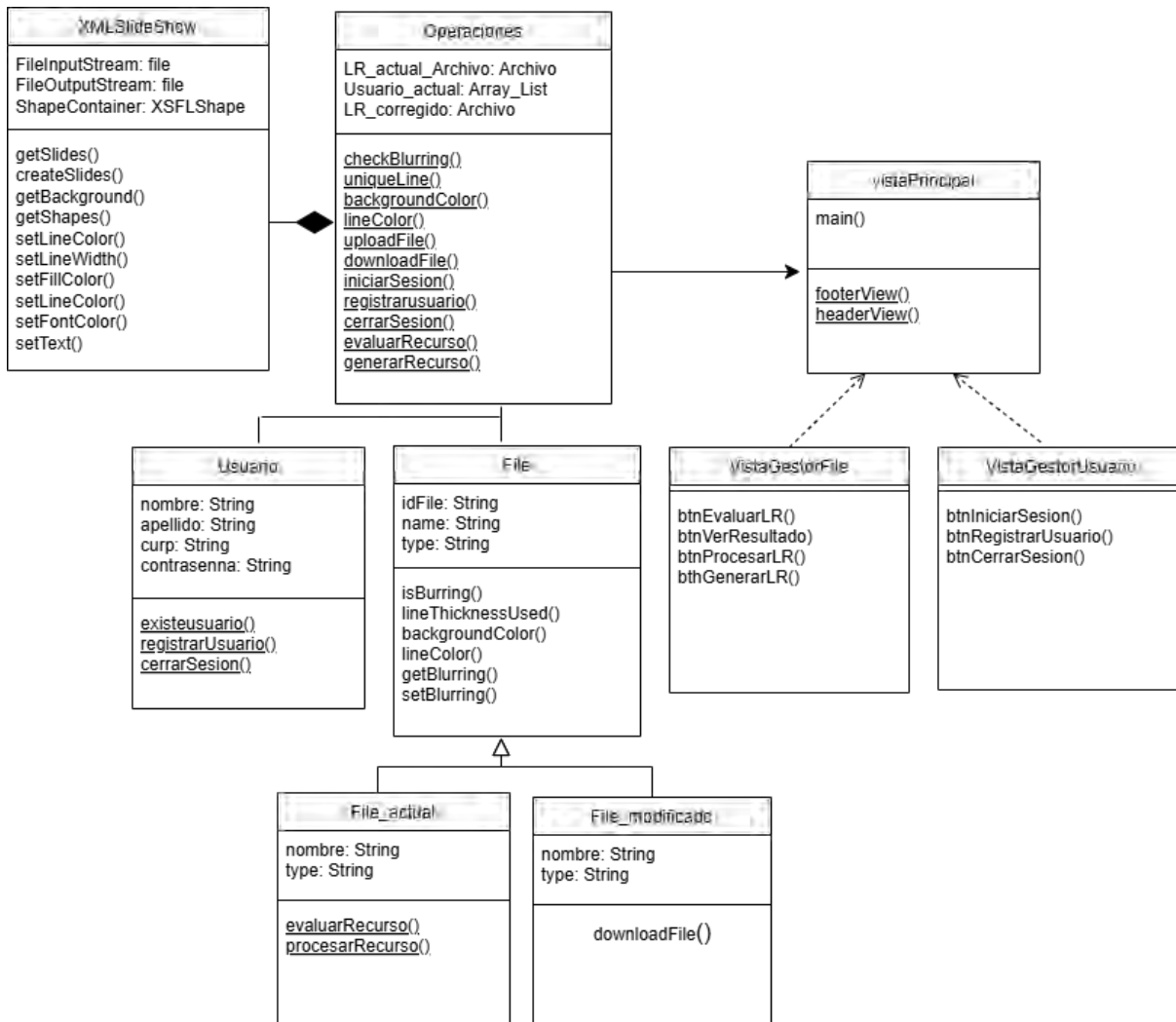


Ilustración 13. Diagrama de Clases de ARE System

4) Descripción general del diagrama de despliegue.

El esquema se muestra en la Ilustración 14.

1. Computadora cliente
 - Contiene el componente “Browser HTML”, encargado de la interacción con el sistema.
2. Servidor
 - TOMCAT 8.0: servidor de aplicaciones que ejecuta la lógica del sistema.
 - Catalina Servlet 3.0: gestiona las solicitudes HTTP y ejecuta el archivo de despliegue are.war.
 - El servidor se comunica con la base de datos utilizando el conector JDBC (Java Database Connectivity)
3. Base de datos (MySQL DB)

- Sistema de almacenamiento que aloja información necesaria para la operación del sistema.

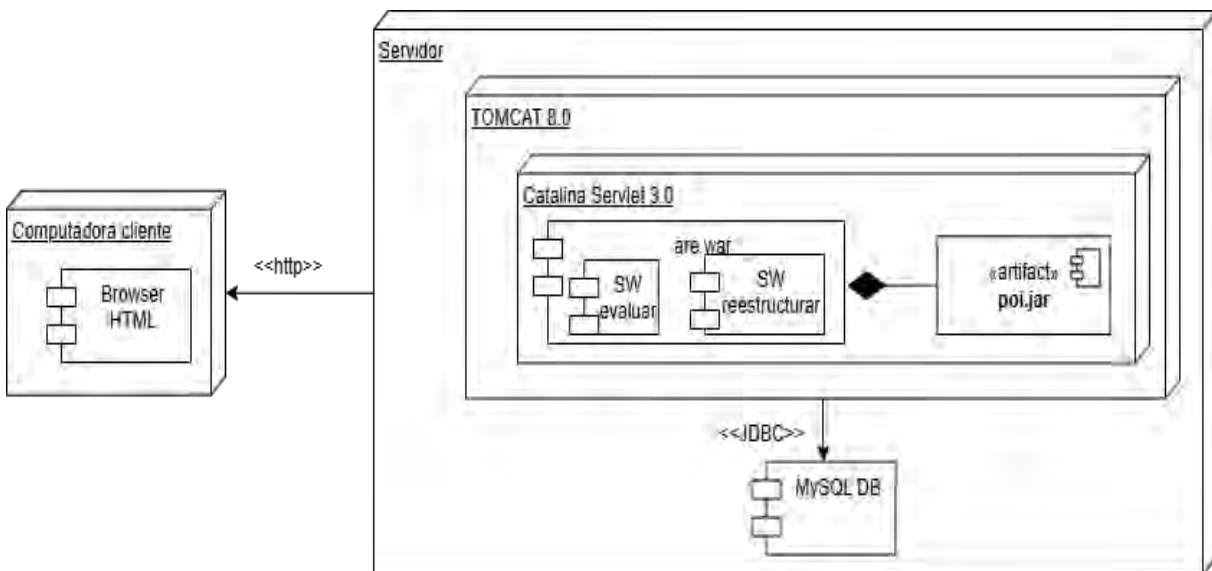


Ilustración 14. Diagrama de Despliegue de ARE System

III.5. Implementación del sistema

En la Ilustración 15 se muestra parte del entorno de desarrollo Eclipse con el proyecto abierto. La herramienta desarrollada emplea tecnologías web, como son el lenguaje HTML, útil para definir la estructura de las páginas web; CSS, que da estilo y formato a los elementos HTML y JavaScript, que ofrece interactividad a las páginas web.

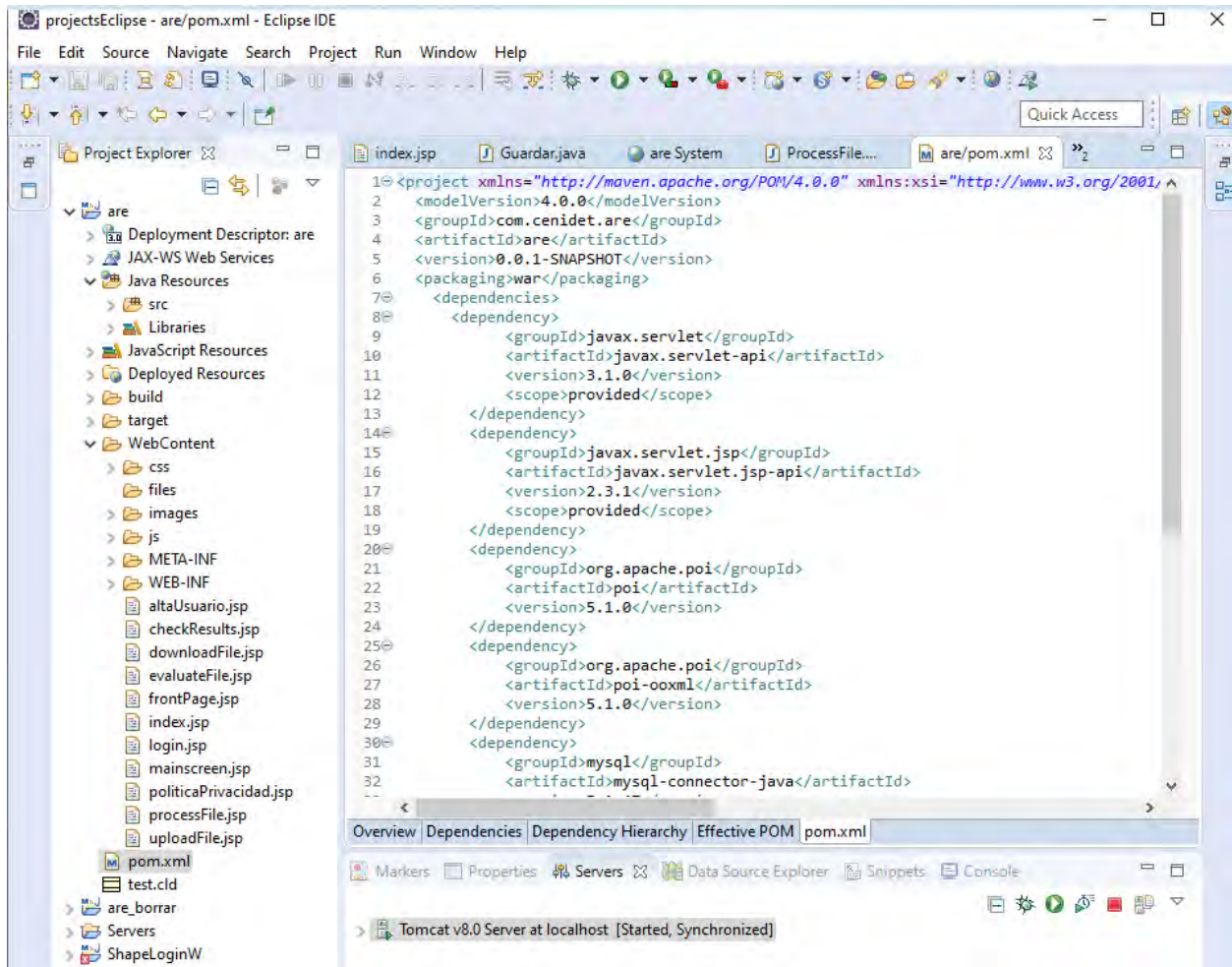


Ilustración 15. Ambiente de desarrollo de trabajo

Además, el proyecto depende de otras bibliotecas para poder manejar archivos e interactuar con bases de datos, ejecutar servlets y JSP. En la Ilustración 16 se muestra parte de la estructura del explorador de proyectos donde se muestra de manera gráfica el uso de esas tecnologías.

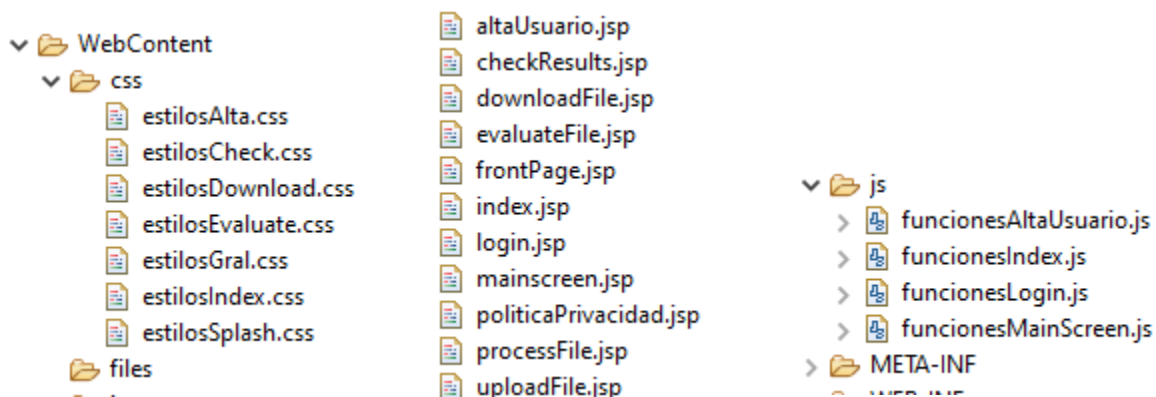


Ilustración 16. Implementación de tecnologías web

La metodología empleada consiste básicamente en leer los metadatos del archivo PowerPoint y a partir de éstos se realiza el proceso de evaluación y reestructura. Se construyó en la plataforma de software de código abierto Eclipse IDE for Enterprise Java Developers Version 2019-09 R (4.13.0) en lenguaje Java, y consta de clases con diversos métodos que permiten realizar tanto la lectura de los metadatos como las modificaciones a los recursos de aprendizaje, apoyándose de la biblioteca Apache POI. El uso de esta biblioteca permite manipular la estructura interna de las presentaciones, extrayendo sus metadatos en formato XML, lo que permite la lectura y escritura de archivos en formato de Microsoft Office.

Para poder trabajar con los recursos de aprendizaje en formato de PowerPoint, Apache POI proporciona la biblioteca XSLF (XML Slide Layout Format). La implementación de XSLF es para archivos PowerPoint en formato XML (Office Open XML .pptx) lo que permite realizar actividades como leer, escribir y modificar este tipo de archivos, facilitando la tarea para extraer y modificar texto dentro de las diapositivas, así como el manejo de estilos, fondos y gráficos. Parte de la implementación se muestra en la Ilustración 17.

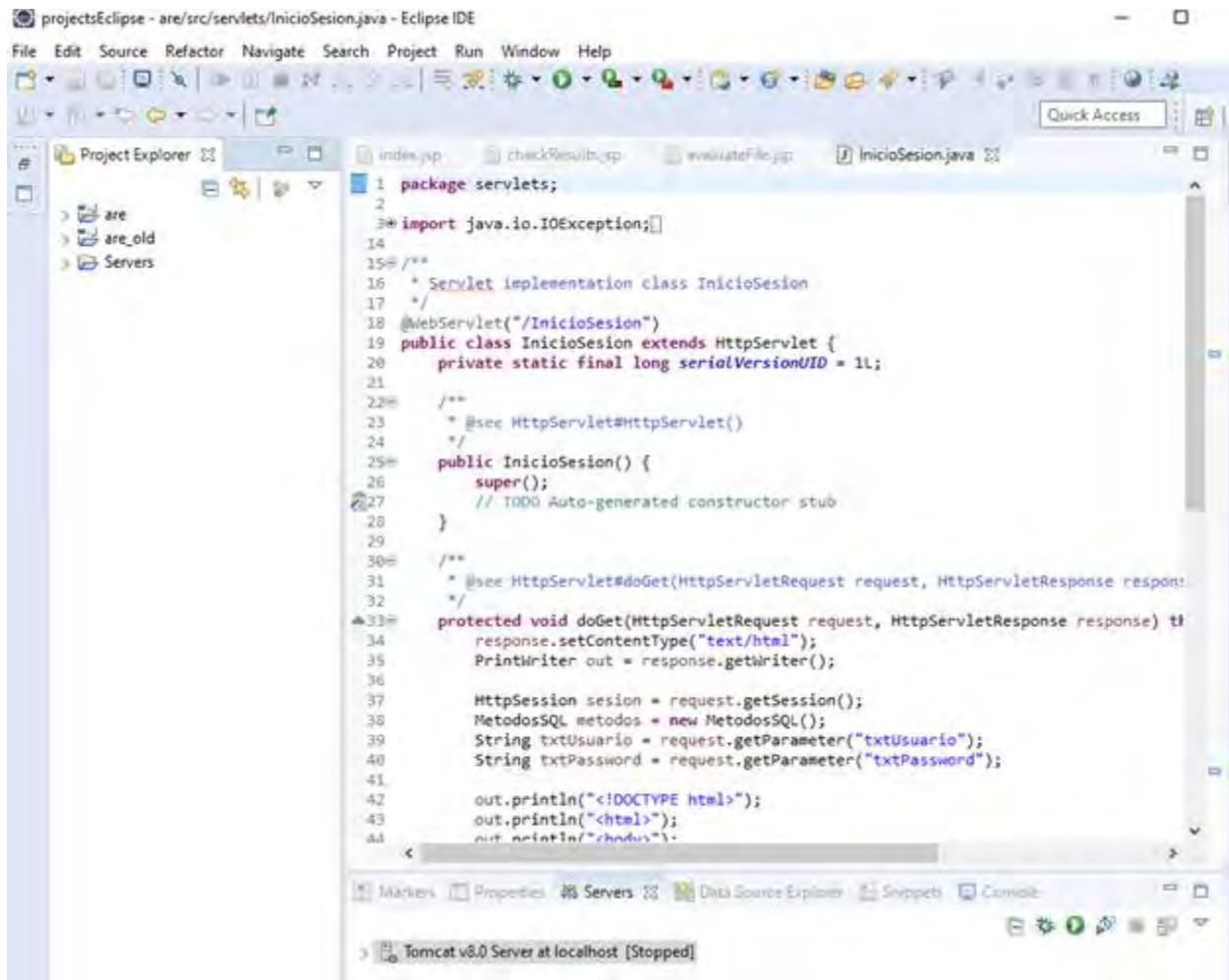


Ilustración 17. Contenido del proyecto are

El sistema se ejecuta sobre un servidor Apache Tomcat v8.0 Server, utilizando MySQL Workbench 8.0 Versión 8.0.34 CE (64 bits) como gestor de bases de datos, el cual se muestra en la Ilustración 18. La base de datos almacena la información referente al usuario, tal como identificador de usuario, contraseña, nombre, apellidos y correo electrónico.. También se hace uso de servlets, estos se encargan de gestionar la sesión de los usuarios, cargar y descargar los recursos de aprendizaje en el sistema y la interacción con la base de datos. Cuando un usuario usa el sistema por vez primera, podrá registrarse, en caso de haberlo hecho con anterioridad, bastará con autenticarse con su identificador de usuario y contraseña. La información del usuario se guarda para fines estadísticos

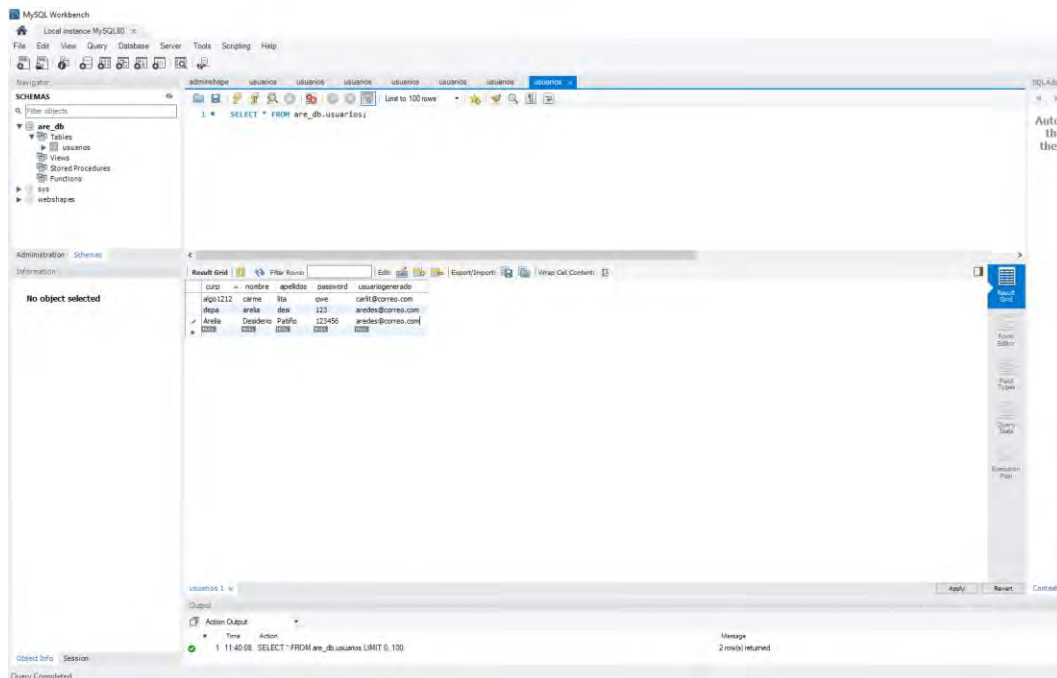


Ilustración 18. Área de trabajo MySQL Workbench

Al momento de ejecutar el sistema, se despliega la página de Bienvenida que se muestra en la Ilustración 19, en ella se proporciona al usuario una breve descripción sobre la funcionalidad del mismo. Lo siguiente que se presenta es la página de Inicio, en ella el usuario puede elegir entre las opciones para darse de alta o iniciar sesión. Cuando un usuario inicia sesión, se le da acceso a todas las funcionalidades del sistema, este proceso requiere que el usuario se haya dado de alta previamente. Sin embargo, cuando un usuario inicia el proceso de registro, implica que solo se hará el alta del usuario, lo que significa que para poder hacer uso del sistema, éste deberá introducir sus credenciales e iniciar sesión de manera habitual. Una vez que el usuario haya ingresado al sistema se desplegará un menú con las funciones principales, donde se muestran acciones asociadas al recurso de aprendizaje, tales como cargar, evaluar, procesar o descargar archivos que contiene dichos recursos de aprendizaje.

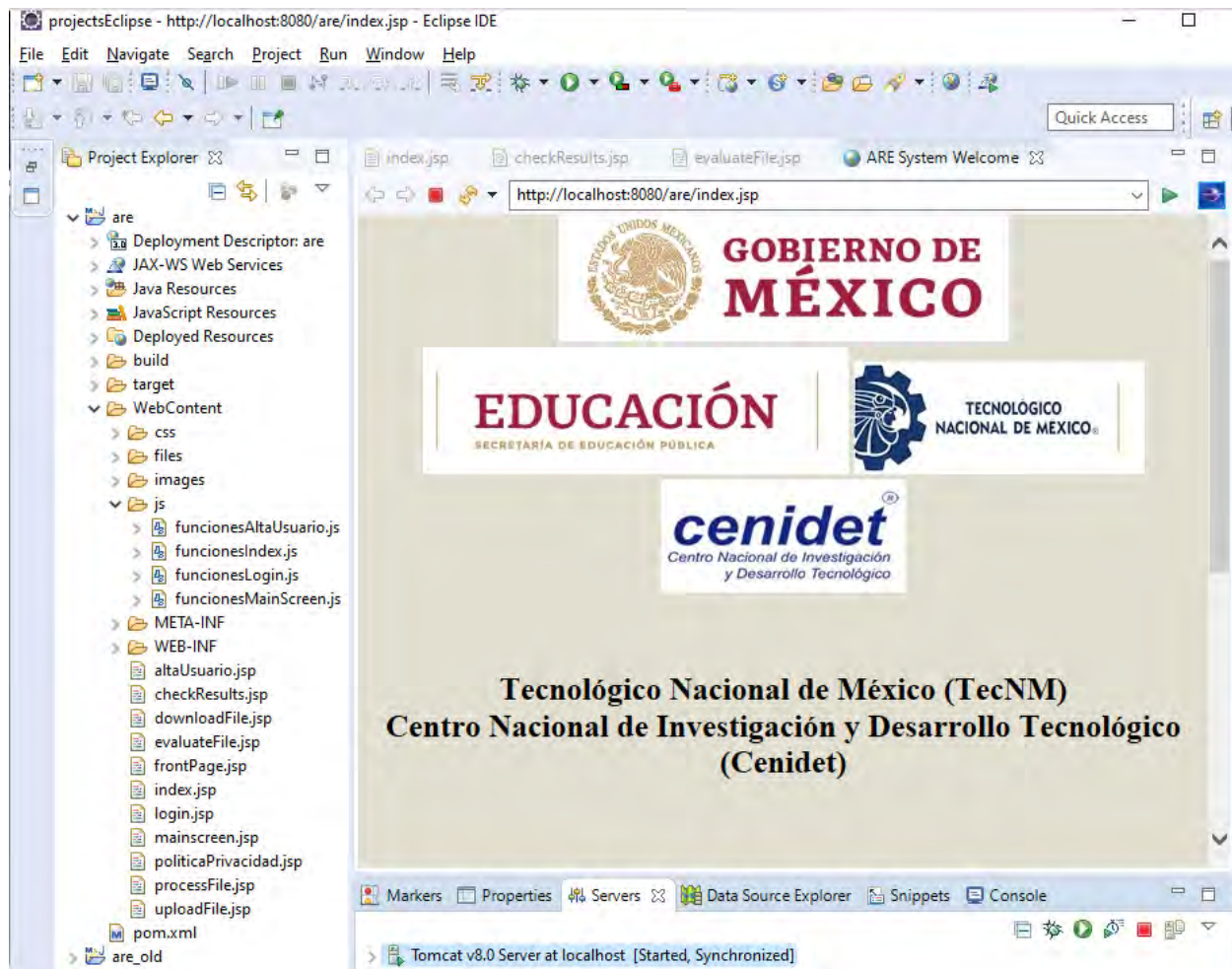


Ilustración 19. Servidor Tomcat en ejecución y sincronizado

Una vez se haya ingresado del sistema, se podrá cargar al servidor los archivos digitales en formato PowerPoint que contienen el recurso de aprendizaje ya sea para evaluar o reestructurar, o ambas, según sea el caso. Tras cargar el archivo, cuando el usuario selecciona la opción evaluar recurso de aprendizaje, el sistema llevará a cabo la evaluación considerando aspectos como, el color y grosor de líneas, el color y tamaño de la letra y el uso de colores de los elementos que componen el diagrama de clases así como del panel de diapositivas. Por otro lado, si se desea reestructurar el recurso de aprendizaje, el usuario podrá elegir esta opción e ir directamente al proceso de reestructura, ya que ambas opciones son independientes, es decir, se puede llevar a cabo el proceso de reestructura sin necesidad de haber realizado la evaluación, o viceversa. Para ello se debe indicar los atributos que se desean modificar marcándolos mediante una casilla de verificación. Si no se selecciona ningún atributo, la herramienta no generará ninguna modificación y en consecuencia no se creará el recurso de aprendizaje reestructurado. Para generar el archivo PowerPoint reestructurado se deberá seleccionar al menos un atributo de la lista de opciones, una vez generado el

recurso este queda a disposición del usuario, ya sea para presentarlo como vista previa o para descargar. De este modo el sistema ofrece al usuario la flexibilidad necesaria para elegir entre las diferentes opciones el parámetro de calidad a reestructurar, permitiendo cambiar el recurso de aprendizaje según sus necesidades. Para finalizar, el usuario podrá elegir entre iniciar un nuevo proceso o simplemente cerrar la sesión.

III.6. Casos de prueba

En esta sección se presentan las pruebas hechas al sistema de acuerdo con las diferentes funcionalidades definidas en la sección III.2, con el fin de comprobar que el producto final satisface las necesidades del usuario. De la Tabla 13 a la Tabla 21 se encuentra la descripción de los Casos de Prueba para cada funcionalidad.

Tabla 13. Descripción del caso de prueba CP001 Iniciar sesión de usuario

Datos generales de la prueba			
ID y Nombre	CP001. Iniciar sesión	Fecha	07/01/2025
Ejecutor	Desarrollador	Cliente	
Arelia Desiderio Patiño	Arelia Desiderio Patiño	Dra. Olivia G. Fragoso Díaz	
Desarrollo			
Objetivo: Validar que la aplicación permite el ingreso al sistema toda vez que el usuario se ha autenticado con sus credenciales.			
Prioridad: Alta			
Id del Requerimiento: RF002			
Requerimiento: El sistema debe permitir autenticar a un usuario mediante sus credenciales (identificador, contraseña).			
Precondiciones: El usuario debe estar dado de alta en el sistema.			
Postcondiciones: Se muestra la página que contiene el menú principal de la aplicación.			
Valores/Datos de entrada: Identificador de usuario y contraseña.			
Resultados esperados: Toda vez que el usuario ha ingresado al sistema se presenta un mensaje de bienvenida.			

Resultados obtenidos: En la Ilustración 20 se muestra el mensaje de bienvenida confirmando que el usuario ha ingresado exitosamente.

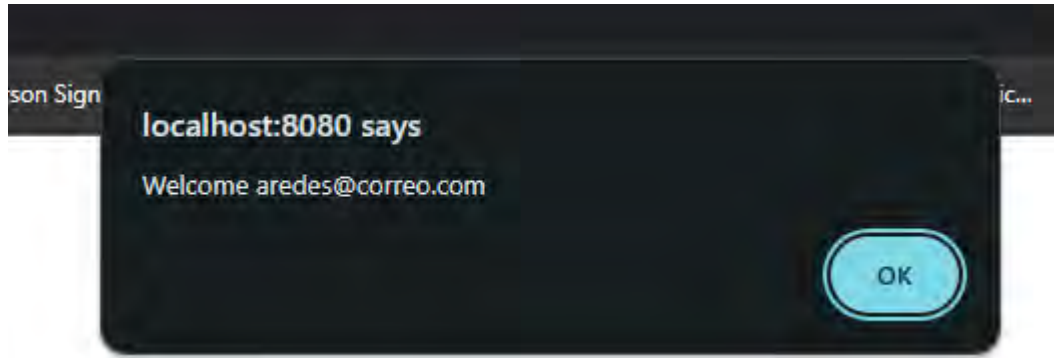


Ilustración 20. Mensaje de confirmación de inicio de sesión exitoso.

Estado de la prueba: Aprobada

Acciones correctivas: Sin observaciones

Aprobación

Arelia Desiderio Patiño

Dra. Olivia G. Fragoso Díaz

Comentarios:

Tabla 14. Descripción del caso de prueba CP001 Iniciar sesión de usuario (Fracaso)

Datos generales de la prueba			
ID y Nombre	CP001. Iniciar sesión	Fecha	07/01/2025
Ejecutor	Desarrollador		Cliente
Arelia Desiderio Patiño	Arelia Desiderio Patiño	Dra. Olivia G. Fragoso Díaz	
Desarrollo			
Objetivo: Validar que la aplicación permite el ingreso al sistema toda vez que el usuario se ha autenticado con sus credenciales.			
Prioridad: Alta			
Id del Requerimiento: RF002			
Requerimiento: El sistema debe permitir autenticar a un usuario mediante sus credenciales (identificador, contraseña).			
Precondiciones: El usuario debe estar dado de alta en el sistema.			
Postcondiciones: Se muestra la página que contiene el menú principal de la aplicación.			
Valores/Datos de entrada: Identificador de usuario y contraseña.			

Resultados esperados: El sistema muestra una notificación indicando al usuario que el ingreso al sistema ha fracasado.

Resultados obtenidos: En la Ilustración 21 se presenta el mensaje informando al usuario que el inicio de sesión no ha sido exitoso.

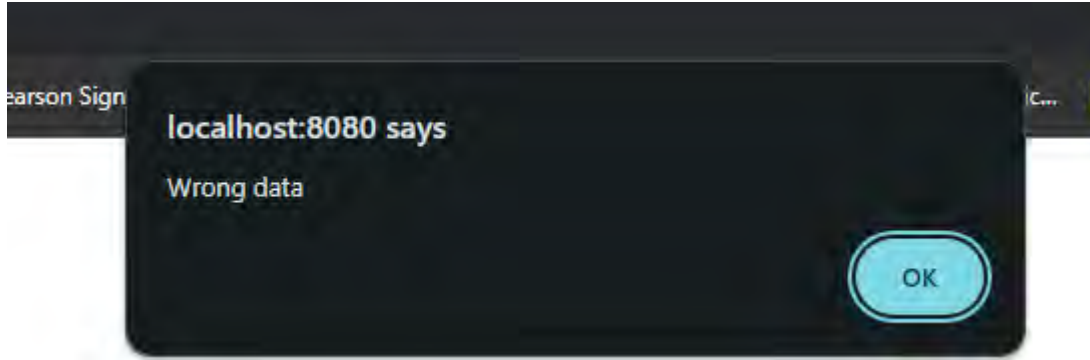


Ilustración 21. Mensaje de inicio de sesión no exitoso.

Estado de la prueba: Aprobada

Acciones correctivas: Sin observaciones

Aprobación

Arelia Desiderio Patiño

Dra. Olivia G. Fragoso Díaz

Comentarios:

Tabla 15. Descripción del caso de prueba CP002. Registrar usuario

Datos generales de la prueba				
ID y Nombre	CP002. Registrar usuario		Fecha	07/01/2025
Ejecutor	Desarrollador		Cliente	
Arelia Desiderio Patiño	Arelia Desiderio Patiño		Dra. Olivia G. Fragoso Díaz	
Desarrollo				
Objetivo: Validar que el sistema registra un nuevo usuario con la información requerida.				
Prioridad: Alta				
Id del Requerimiento: RF006				
Requerimiento: El sistema debe permitir a nuevos usuarios registrarse capturando los siguientes datos: CURP, nombre, apellido y clave.				
Precondiciones: El sistema muestra el formulario donde el usuario deberá ingresar los datos solicitados.				

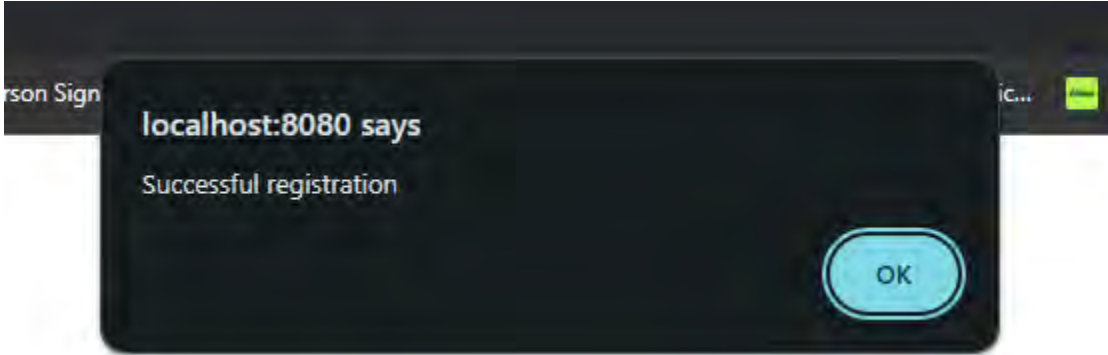
Postcondiciones: El sistema muestra la página de inicio en espera de alguna solicitud del nuevo usuario.	
Valores/Datos de entrada: CURP, nombre, apellido y clave.	
Resultados esperados: Toda vez que el usuario se ha dado de alta, se presenta la página de inicio en espera de alguna solicitud del nuevo usuario.	
Resultados obtenidos: En la Ilustración 22 se muestra el mensaje de confirmación una vez finalizado el proceso de alta de usuario.	
 Ilustración 22. Mensaje de confirmación de registro de usuario exitoso	
Estado de la prueba: Aprobada	
Acciones correctivas: Sin observaciones	
Aprobación	
_____ Arelia Desiderio Patiño	_____ Dra. Olivia G. Fragoso Díaz
Comentarios:	

Tabla 16. Descripción del caso de prueba CP002. Registrar usuario (Fracaso)

Datos generales de la prueba			
ID y Nombre	CP002. Registrar usuario	Fecha	07/01/2025
Ejecutor	Desarrollador	Cliente	
Arelia Desiderio Patiño	Arelia Desiderio Patiño	Dra. Olivia G. Fragoso Díaz	
Desarrollo			
Objetivo: Validar el registro de un nuevo usuario con la información requerida.			
Prioridad: Alta			
Id del Requerimiento: RF006			
Requerimiento: El sistema debe permitir a nuevos usuarios registrarse capturando los siguientes datos: CURP, nombre, apellido y clave.			

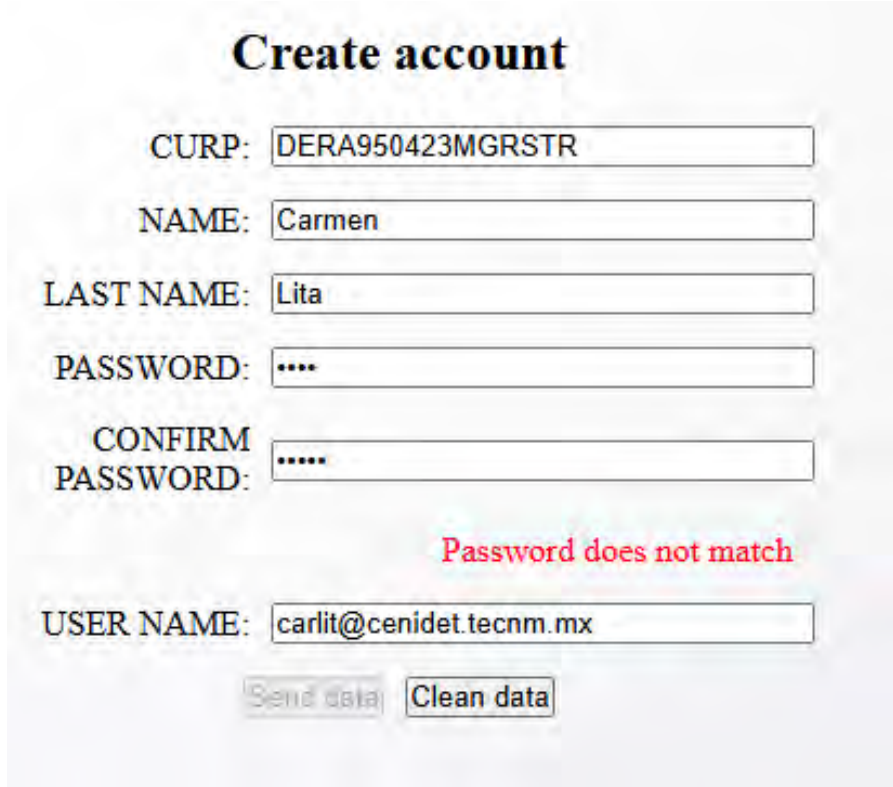
Precondiciones: El sistema muestra el formulario donde el usuario deberá ingresar los datos solicitados.	
Postcondiciones: El sistema muestra la página de inicio en espera de alguna solicitud del nuevo usuario.	
Valores/Datos de entrada: CURP, nombre, apellido y clave.	
Resultados esperados: El sistema no puede validar los datos introducidos y en consecuencia el usuario no podrá registrarse en el sistema.	
Resultados obtenidos: En la captura de pantalla que se presenta en la Ilustración 23 se muestra el formulario para dar de alta nuevos usuarios, solicitando corregir la información proporcionada.	
 Ilustración 23. Formulario de registro de usuarios.	
Estado de la prueba: Aprobada	
Acciones correctivas: Sin observaciones	
Aprobación	
_____ Arelia Desiderio Patiño	_____ Dra. Olivia G. Fragoso Díaz
Comentarios:	

Tabla 17. Descripción del caso de prueba CP003. Cargar archivo

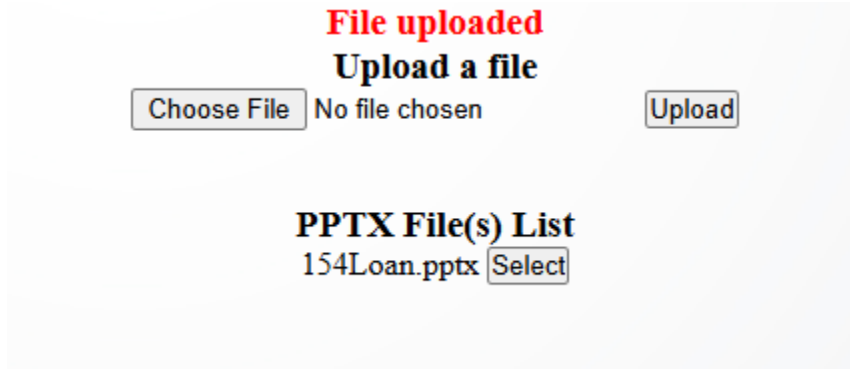
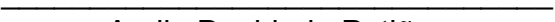
Datos generales de la prueba			
ID y Nombre	CP003. Cargar archivo	Fecha	07/01/2025
Ejecutor	Desarrollador	Cliente	
Areliá Desiderio Patiño	Areliá Desiderio Patiño	Dra. Olivia G. Fragoso Díaz	
Desarrollo			
Objetivo: Validar el proceso de carga de recursos de aprendizaje al sistema.			
Prioridad: Alta			
Id del Requerimiento: RF010			
Requerimiento: El sistema debe permitir al usuario cargar el archivo que contiene el recurso de aprendizaje			
Precondiciones: Tener el recurso de aprendizaje en formato PowerPoint disponible.			
Postcondiciones: El sistema queda cargado con el archivo PowerPoint que contiene el recurso de aprendizaje.			
Valores/Datos de entrada: Archivo en formato PowerPoint que contiene el recurso de aprendizaje.			
Resultados esperados: El sistema queda preparado con el recurso de aprendizaje. Se envía una notificación de carga exitosa.			
Resultados obtenidos: En la Ilustración 24 se presenta la recepción del archivo PowerPoint que contiene el recurso de aprendizaje, listo para procesar.			
 Ilustración 24. Subida de archivos al sistema.			
Estado de la prueba: Aprobada			
Acciones correctivas: Sin observaciones			
Aprobación			
 Areliá Desiderio Patiño		 Dra. Olivia G. Fragoso Díaz	
Comentarios:			

Tabla 18. Descripción del caso de prueba CP004 Evaluar recurso

Datos generales de la prueba				
ID y Nombre	CP004. Evaluar recurso		Fecha	07/01/2025
Ejecutor	Desarrollador		Cliente	
Arelia Desiderio Patiño	Arelia Desiderio Patiño		Dra. Olivia G. Fragoso Díaz	
Desarrollo				
Objetivo: Validar que el sistema analiza los atributos de calidad asociados al recurso de aprendizaje en formato PowerPoint.				
Prioridad: Alta				
Id del Requerimiento: RF013				
Requerimiento: El sistema debe llevar a cabo las tareas correspondientes al proceso de evaluación del recurso de aprendizaje en formato PowerPoint.				
Precondiciones: El sistema lee los metadatos del recurso de aprendizaje en formato PowerPoint.				
Postcondiciones: El sistema muestra un conjunto de valores como resultado del proceso de evaluación asociados a los atributos de calidad del recurso de aprendizaje en formato PowerPoint.				
Valores/Datos de entrada: La lista de los metadatos del archivo que contiene el recurso de aprendizaje.				
Resultados esperados: El sistema muestra los porcentajes de evaluación del recurso de aprendizaje.				
Resultados obtenidos: En la siguiente captura de pantalla que se muestra en la Ilustración 25, se presenta el resultado del proceso de evaluación del recurso de aprendizaje.				

Evaluation Results		
Weight	Criteria	Check
7%	Slide Background	☐
14%	Line Color	☒
10%	Line Width	☐
8%	Shape width	☐
7%	Shape Background	☐
10%	Font Size	☐
10%	Font Color	☐
34%	Not Blurring	☒

The score is 48%

Estado de la prueba: Aprobada

Acciones correctivas: Sin observaciones

Aprobación

_____	_____
Arelia Desiderio Patiño	Dra. Olivia G. Fragoso Díaz

Comentarios:

Ilustración 25. Resultado del proceso de evaluación de un recurso de aprendizaje.

Tabla 19. Descripción del caso de prueba CP005 Procesar recurso

Datos generales de la prueba				
ID y Nombre	CP005. Procesar recurso		Fecha	07/01/2025
Ejecutor	Desarrollador	Cliente		
Arelia Desiderio Patiño	Arelia Desiderio Patiño	Dra. Olivia G. Fragoso Díaz		
Desarrollo				
Objetivo: Validar que el sistema modifica el recurso de aprendizaje en formato PowerPoint con base en la selección de los atributos de calidad.				
Prioridad: Alta				
Id del Requerimiento: RF016				
Requerimiento: El sistema debe realizar las modificaciones al recurso de aprendizaje con base en la lista de atributos de calidad seleccionados por el usuario.				
Precondiciones: El sistema recibe la lista de los atributos de calidad seleccionados				

por el usuario, los cuales determinan el proceso de reestructura.

Postcondiciones: El sistema queda listo para atender la siguiente instrucción.

Valores/Datos de entrada: La lista de los atributos de calidad seleccionados por el usuario.

Resultados esperados: El recurso de aprendizaje se actualiza de acuerdo con los atributos de calidad indicados por el usuario.

Resultados obtenidos: En la siguiente captura de pantalla que se muestra en la Ilustración 26, se visualiza el resultado del proceso de reestructura del recurso de aprendizaje.

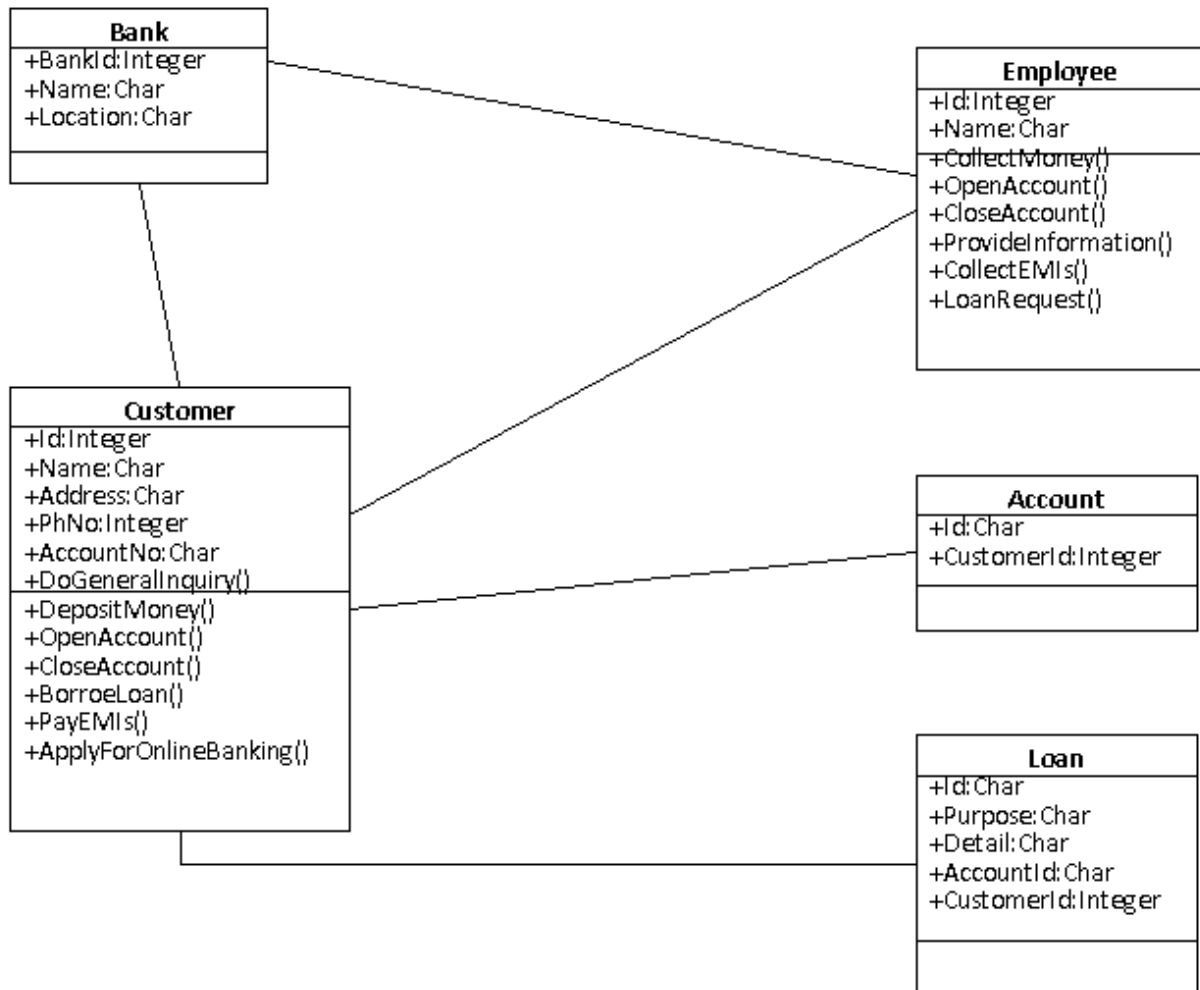


Ilustración 26. Recurso de aprendizaje reestructurado.

Estado de la prueba: Aprobada

Acciones correctivas: Sin observaciones

Aprobación

Arelia Desiderio Patiño	Dra. Olivia G. Fragoso Díaz
Comentarios:	

Tabla 20. Descripción del caso de prueba CP006. Generar recurso

Datos generales de la prueba				
ID y Nombre	CP006. Generar recurso		Fecha	07/01/2025
Ejecutor	Desarrollador		Cliente	
Arelia Desiderio Patiño	Arelia Desiderio Patiño		Dra. Olivia G. Fragoso Díaz	
Desarrollo				
Objetivo: Validar que el sistema genera un recurso de aprendizaje reestructurado en formato PowerPoint.				
Prioridad: Alta				
Id del Requerimiento: RF017				
Requerimiento: El sistema debe generar un nuevo recurso de aprendizaje en formato PowerPoint de acuerdo con los criterios de calidad seleccionados por el usuario.				
Precondiciones: El sistema debe recibir el recurso de aprendizaje con la lista de opciones de los atributos de calidad que modificará el recurso de aprendizaje.				
Postcondiciones: El sistema debe mostrar al usuario el recurso de aprendizaje reestructurado en formato PowerPoint.				
Valores/Datos de entrada: El recurso de aprendizaje modificado según la lista de los atributos de calidad seleccionados por el usuario.				
Resultados esperados: El sistema debe generar el nuevo recurso de aprendizaje reestructurado.				
Resultados obtenidos: En la captura de pantalla de la Ilustración 27 se presenta el recurso de aprendizaje reestructurado listo para descargar.				

Press button below to download the resource shown below

File selected: fileprocessed.pptx

[Download](#)

```

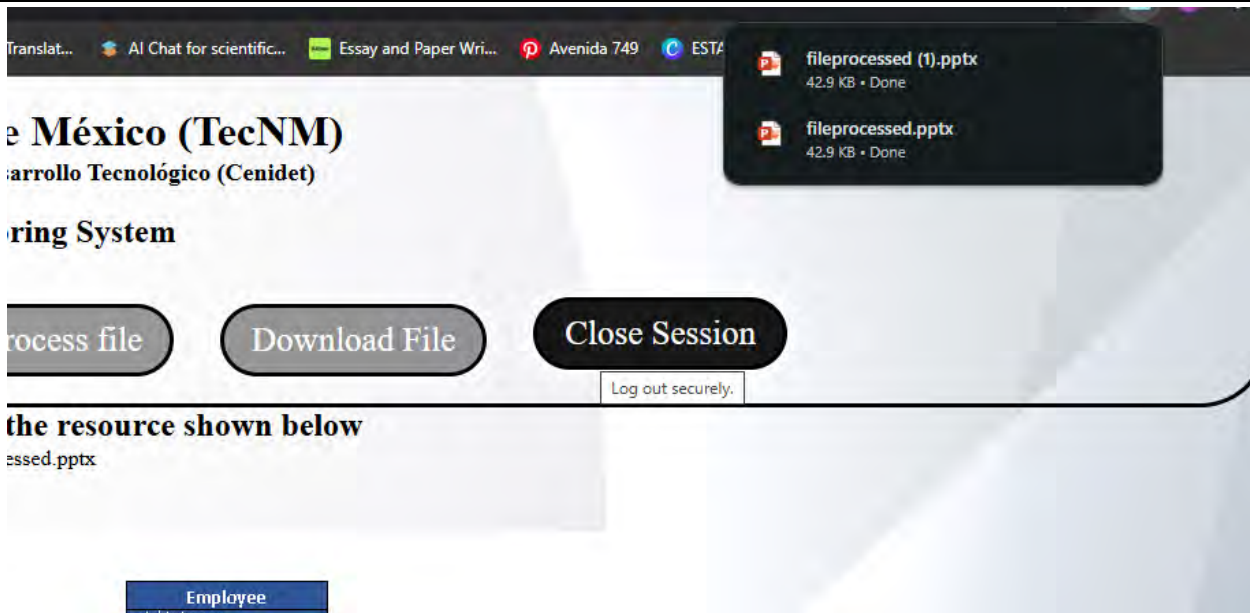
classDiagram
    class Bank {
        +BankId: Integer
        +Name: Char
        +Location: Char
    }
    class Employee {
        +Id: Integer
        +Name: Char
        +Collectivemoney()
        +OpenAccount()
        +CloseAccount()
        +ProvideInformation()
        +CollectEMIs()
        +LoanRequest()
    }
    class Customer {
        +Id: Integer
        +Name: Char
        +Address: Char
        +PhNo: Integer
        +AccountNo: Char
        +DoGeneralInquiry()
        +DepositMoney()
        +OpenAccount()
        +CloseAccount()
        +BorrowLoan()
        +PayEMIs()
        +ApplyForOnlineBanking()
    }
    class Account {
        +Id: Char
        +CustomerId: Integer
    }
    class Loan {
        +Id: Char
        +Purpose: Char
        +Detail: Char
        +AccountId: Char
        +CustomerId: Integer
    }
    Bank -- Employee
    Bank -- Customer
    Employee -- Customer
    Customer -- Account
    Customer -- Loan
    
```

Ilustración 27. Archivo generado y listo para descargar.

Estado de la prueba: Aprobada			
Acciones correctivas: Sin observaciones			
Aprobación			
_____ Arelia Desiderio Patiño		_____ Dra. Olivia G. Fragoso Díaz	
Comentarios:			

Tabla 21. Descripción del caso de prueba CP007. Descargar recurso

Datos generales de la prueba			
ID y Nombre	CP007. Descargar	Fecha	07/01/2025

	recurso		
Ejecutor	Desarrollador	Cliente	
Areliá Desiderio Patiño	Areliá Desiderio Patiño	Dra. Olivia G. Fragoso Díaz	
Desarrollo			
Objetivo: Validar que el sistema permite descargar el recurso de aprendizaje en formato PowerPoint.			
Prioridad: Alta			
Id del Requerimiento: RF019			
Requerimiento: El sistema debe permitir descargar el archivo con los cambios aplicados en formato PowerPoint.			
Precondiciones: El sistema recibe la instrucción para exportar el archivo con el recurso de aprendizaje reestructurado.			
Postcondiciones: El sistema queda preparado para atender la nueva solicitud del usuario.			
Valores/Datos de entrada: El archivo reestructurado que contiene el recurso de aprendizaje en formato PowerPoint.			
Resultados esperados: El sistema muestra la confirmación indicando que la descarga del recurso de aprendizaje ha sido exitosa.			
Resultados obtenidos: El mensaje de la Ilustración 28 muestra la finalización del proceso de descarga del recurso de aprendizaje reestructurado.			
			
Ilustración 28. Confirmación de descarga del recurso de aprendizaje.			
Estado de la prueba: Aprobada			

Acciones correctivas: Sin observaciones	
Aprobación	
Areliá Desiderio Patiño	Dra. Olivia G. Fragoso Díaz
Comentarios:	

Capítulo IV

Análisis de resultados

Este capítulo describe los resultados de las pruebas realizadas con los diferentes recursos de aprendizaje, al final se incluyen algunas aportaciones de los resultados obtenidos.

IV. Análisis de resultados de pruebas

De acuerdo con la tesis de (Sandoval, 2023) se cuenta con 114 casos de estudio de recursos de aprendizaje tomados de internet, con los cuales se han hecho varias pruebas al sistema, sin embargo, por razones de espacio solo se presenta un caso de estudio al cual se aplican diferentes procesos de reestructuración. En la Ilustración 29 se muestra el recurso de aprendizaje “154Loan.pptx”, el cual se toma como ejemplo porque no cumple con ciertas características consideradas en el Modelo de Correctitud, como son el uso de colores distintos a los recomendados dentro de las clases, así como en el panel de diapositivas y el texto. El utilizar colores no adecuados puede provocar la dificultad en la claridad visual del diagrama y como consecuencia, puede comprometer su efectividad como recurso de aprendizaje. Además, para mostrar las diferentes capacidades del sistema en el proceso de reestructura, se ha hecho una copia del archivo original y se han alterado algunos elementos, como son el tamaño del texto y el grosor de las líneas.

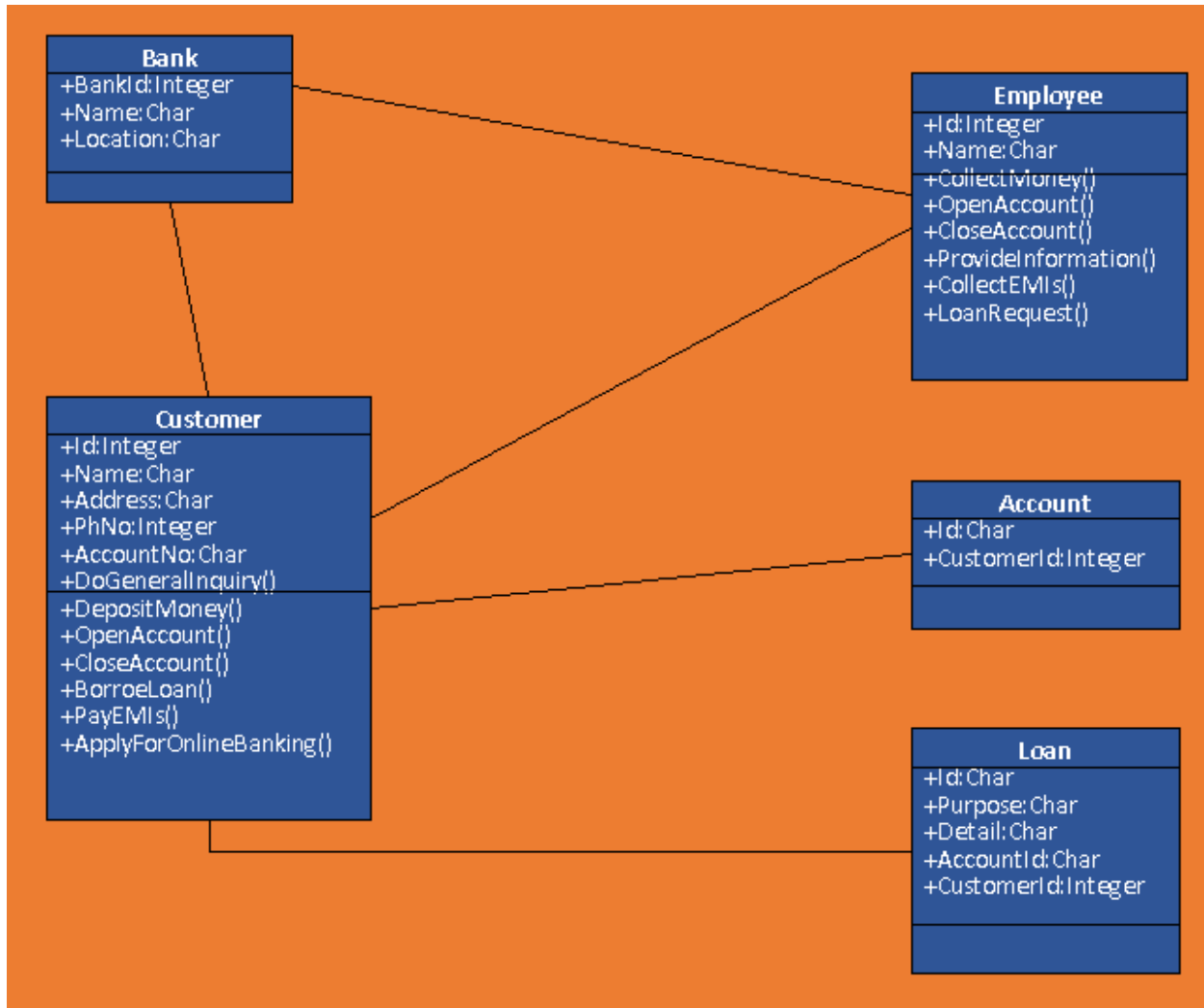


Ilustración 29. Recurso de aprendizaje “154Loan.pptx”

Al someter el recurso de aprendizaje “154Loan.pptx” al proceso de evaluación de ARE System, se obtuvo una valoración inicial de 48% de nivel de correctitud. Esta puntuación se obtiene al sumar los pesos de cada uno de los atributos que si cumple los requerimientos, estos pesos se indican del lado izquierdo de cada atributo y el resultado final se muestra en la Ilustración 30. Aquellos atributos que se muestran del lado derecho con una marca en la casilla de verificación, son los que sí cumplen con las especificaciones de evaluación, mientras que aquellos que no tienen la marca indica el incumplimiento de ese atributo. A continuación se muestran los resultados obtenidos de cada uno de los atributos evaluados.

Evaluation Results		
Weight	Criteria	Check
7%	Slide Background	☐
14%	Line Color	☒
10%	Line Width	☐
8%	Shape width	☐
7%	Shape Background	☐
10%	Font Size	☐
10%	Font Color	☐
34%	Blurring	☒
The score is 48%		

Ilustración 30. Nivel de correctitud del recurso de aprendizaje.

a. Text color

Este atributo se asocia al color del texto y la primera prueba consiste en reestructurar dicho atributo siguiendo las reglas del Modelo de Correctitud. En ocasiones ocurre que el autor del recurso de aprendizaje utiliza colores que no contrastan con el color de fondo. Para validar esta prueba el sistema debe ser capaz de modificar el color de fuente. El resultado que se obtiene al reestructurar este atributo se muestra en la Ilustración 31, en la cual se aprecia el cambio de color de fuente de blanco a negro. De acuerdo con el resultado, se tiene un diagrama de clases donde se ha modificado el color del texto y el nivel de correctitud ha sido mejorado de un 48% a un 58%. El resultado de la evaluación se muestra en la Ilustración 32.

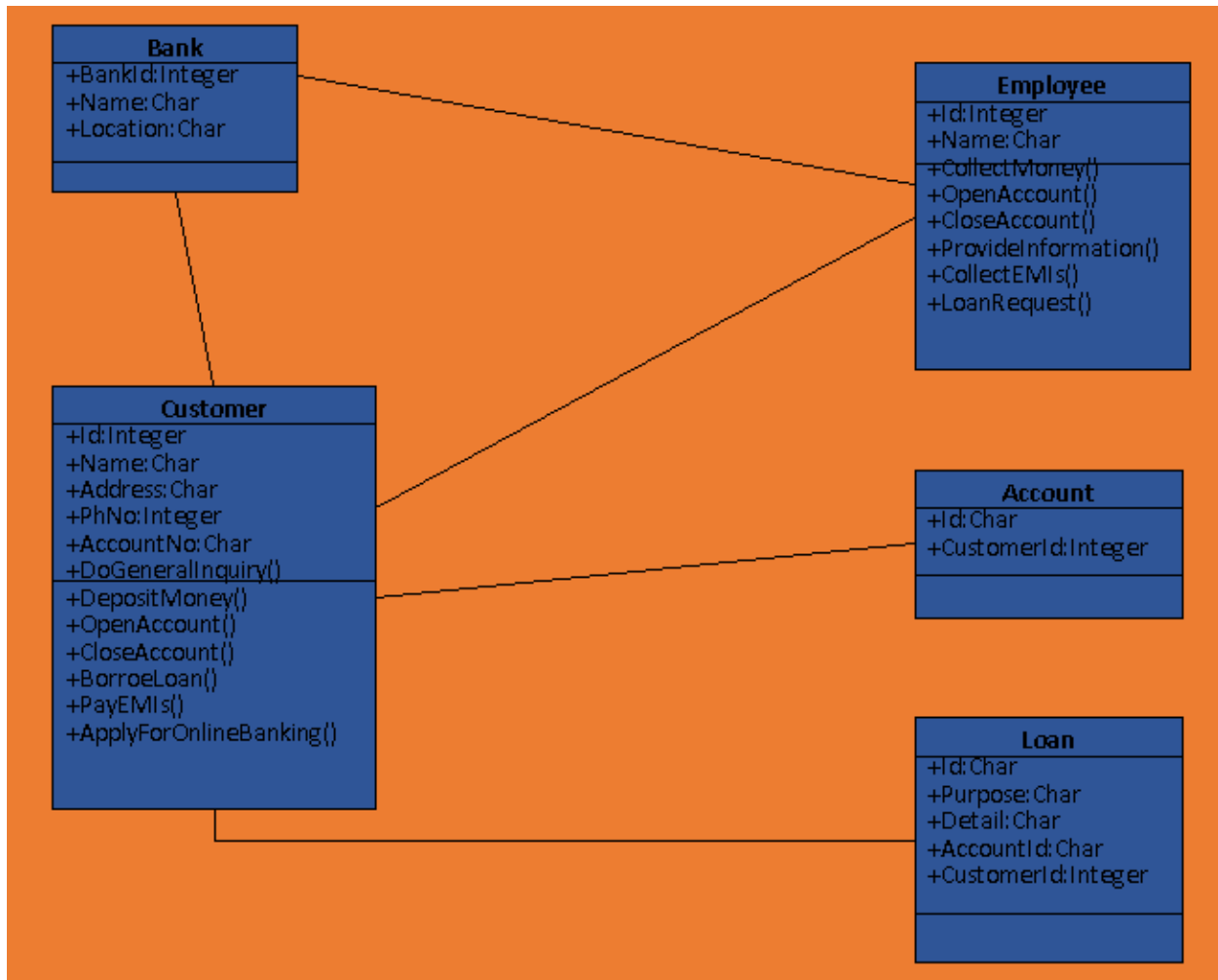


Ilustración 31. Recurso de aprendizaje reestructurado con la opción Text color

Evaluation Results		
Weight	Criteria	Check
7%	Slide Background	☐
14%	Line Color	☒
10%	Line Width	☐
8%	Shape width	☐
7%	Shape Background	☐
10%	Font Size	☐
10%	Font Color	☒
34%	Blurring	☒
The score is 58%		

Ilustración 32. Nivel de correctitud después de cambiar el color del texto.

b. Slide Background

Este atributo se refiere al uso de colores en el fondo del panel de diapositivas. Tomando como referencia el ejemplo 154Loan.pptx, el cual se observa con fondo de color naranja, se requiere cambiar a un fondo blanco. En la Ilustración 33 se muestra el resultado de modificar el color de fondo de la diapositiva. Para ello se debe reestructurar el atributo Slide Background. Este cambio solo afecta al color de fondo de la diapositiva, para cambiar el color de clases se deberá usar otra opción, la cual se muestra a continuación.

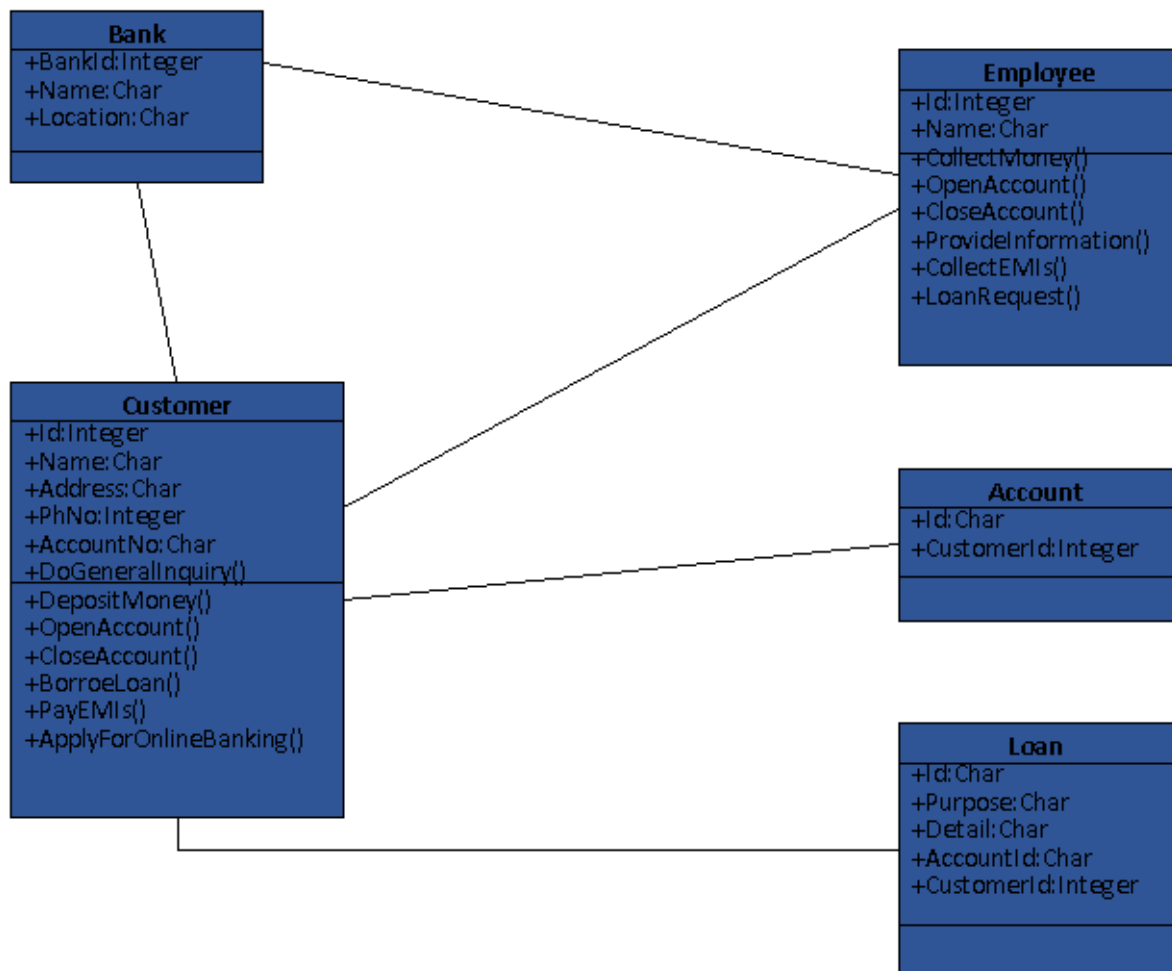


Ilustración 33. Recurso de aprendizaje reestructurado con la opción Slide Background.

En este caso, los cambios del color de fondo de la diapositiva de color naranja a blanco facilita significativamente la claridad visual del diagrama, lo que contribuye a mejorar la calidad visual del recurso de aprendizaje. Al evaluar este recurso con los cambios

aplicados, en la Ilustración 34 se observa un aumento en el nivel de correctitud, es decir, se ha cambiado el color de fuente y el color del panel de diapositivas, lo que representa un 65% de nivel de correctitud.

Evaluation Results		
Weight	Criteria	Check
7%	Slide Background	☒
14%	Line Color	☒
10%	Line Width	☐
8%	Shape width	☐
7%	Shape Background	☐
10%	Font Size	☐
10%	Font Color	☒
34%	Blurring	☒
The score is 65%		

Ilustración 34. Nivel de correctitud después de cambiar el color del panel de diapositivas.

c. Shape Background

Otra de las pruebas consiste en cambiar el color de fondo de los clasificadores de clases, para ello se aplica la opción Shape Background. El Modelo de Correctitud no recomienda el uso de colores dentro de los diagramas, por lo que esta prueba se debe aplicar cuando el color de fondo de las clases sea diferente del color blanco, esto con el fin de evitar el uso excesivo de colores que puedan distraer la atención del usuario. Cabe mencionar que los cambios realizados en los ejemplos muestran la capacidad del sistema para reestructurar ya sea en conjunto o sólo aquellos atributos que han sido seleccionados.

En la siguiente prueba se desea eliminar el uso de colores en el fondo de las clases, en el diagrama original las clases se representan con un color azul, por lo que se desea cambiar a un fondo blanco, el resultado se muestra en la Ilustración 35. Anteriormente se mencionó que el sistema permite modificar tanto el color del panel de diapositivas como el color de fondo de las clases, en este caso el resultado que se muestra corresponde al cambio de color en el fondo de las clases.

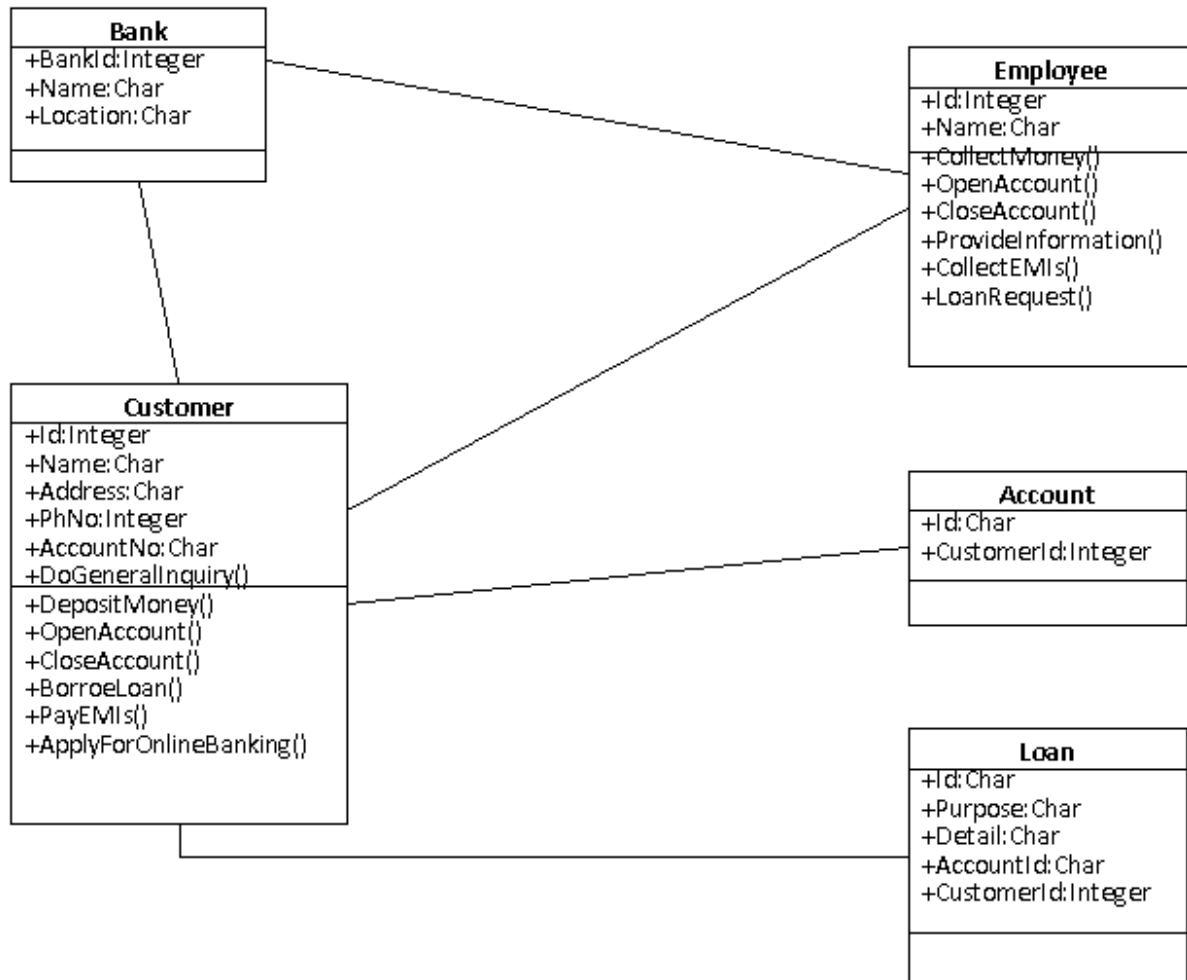


Ilustración 35. Recurso de aprendizaje reestructurado con la opción Shape Background.

Al momento de evaluar el recurso de aprendizaje con las modificaciones realizadas hasta ahora, se tiene un nivel de correctitud del 72%, los resultados de la evaluación se presentan en la Ilustración 36.

Evaluation Results		
Weight	Criteria	Check
7%	Slide Background	☒
14%	Line Color	☒
10%	Line Width	☐
8%	Shape width	☐
7%	Shape Background	☒
10%	Font Size	☐
10%	Font Color	☒
34%	Blurring	☒
The score is 72%		

Ilustración 36. Nivel de correctitud después de cambiar el color del fondo de las clases.

d. Font size

Los cambios aplicados anteriormente muestran el procedimiento para modificar el color de fuente, sin embargo, otro de los atributos considerados en el Modelo de Correctitud corresponde al tamaño de fuente y para ello se debe aplicar la opción Font size, la cual recomienda no usar textos demasiados grandes ni demasiado pequeños dentro de las diapositivas. Dentro del Modelo de Correctitud se menciona que el tamaño de fuente de 12 pixeles se considera el apropiado, sin embargo, esto es preferible cuando se trata de documentos, por lo que en este trabajo se sugiere el tamaño de fuente de 14 pixeles considerándolo el adecuado. De esta manera también se previene el hecho de que al modificar el tamaño del texto no de una apariencia desproporcionada en relación con el tamaño del objeto o la figura que lo contiene. Una vez que se ha modificado el tamaño de fuente, el resultado se presenta en la Ilustración 37.

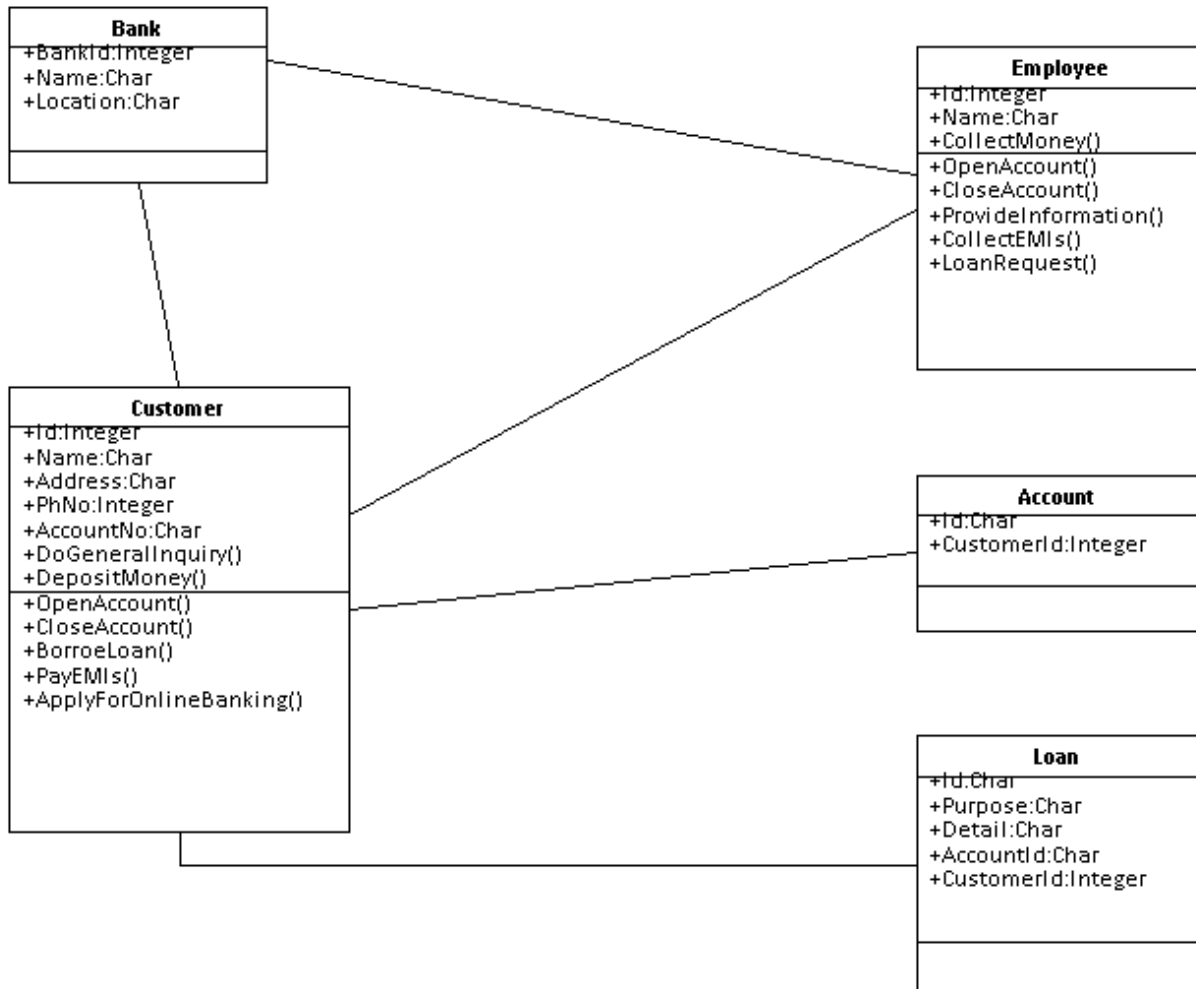


Ilustración 37. Recurso de aprendizaje reestructurado con la opción Font size.

Una vez que se han aplicado los cambios, el recurso de aprendizaje se evalúa nuevamente, el valor que se obtiene corresponde a un 82% de correctitud y el resultado se muestra en la Ilustración 38.

Evaluation Results		
Weight	Criteria	Check
7%	Slide Background	☒
14%	Line Color	☒
10%	Line Width	☐
8%	Shape width	☐
7%	Shape Background	☒
10%	Font Size	☒
10%	Font Color	☒
34%	Blurring	☒
The score is 82%		

Ilustración 38. Nivel de correctitud después de cambiar el tamaño de fuente.

e. *Shape width, Line width*

Para finalizar, se eligen las opciones que modifican el grosor de las líneas. Dentro de este atributo se consideran dos aspectos, por un lado se reestructura el grosor de las líneas de los rectángulos que representan las clases y por otro, el grosor de las líneas que representan las relaciones entre clases, con ello se ofrece al usuario la libertad de modificar ambos atributos o por separado. Entonces, el grosor de las líneas se considera como atributo dentro de un diagrama de clases, debido a que las líneas demasiado gruesas podrían dificultar el seguimiento de estas, en este sentido, la recomendación es evitar el uso de líneas demasiado finas pero tampoco que sean demasiado gruesas, también tratar que las líneas no sean demasiado largas pero tampoco demasiado cortas, y por último, se recomienda el uso de líneas rectas y ortogonales (Dabo, 2005). Para este caso, al momento de evaluar el recurso de aprendizaje los resultados muestran que el ancho de líneas no corresponden con los parámetros recomendados, lo que sugiere que se deben reestructurar. Con el fin de mostrar al lector que se pueden aplicar los cambios en conjunto, en este ejemplo se aplican ambas acciones. Los resultados después de realizar los cambios se muestran en la Ilustración 39 y el resultado de evaluación se observa en la Ilustración 40.

f. *Resultado final*

Los cambios descritos en el diagrama de clases usado como ejemplo reflejan el proceso de reestructura de los diferentes atributos. Es importante mencionar que al

momento de presentar el resultado en los diagramas se aprecia como los elementos parecen superpuestos unos con otros, siendo este un aspecto propio de la tecnología utilizada, sin embargo, al momento de descargar el recurso de aprendizaje reestructurado este efecto no se presenta más. En la Ilustración 39 se visualiza el resultado final del proceso de reestructura del recurso de aprendizaje que ha sido descargado del sistema con todos los cambios aplicados.

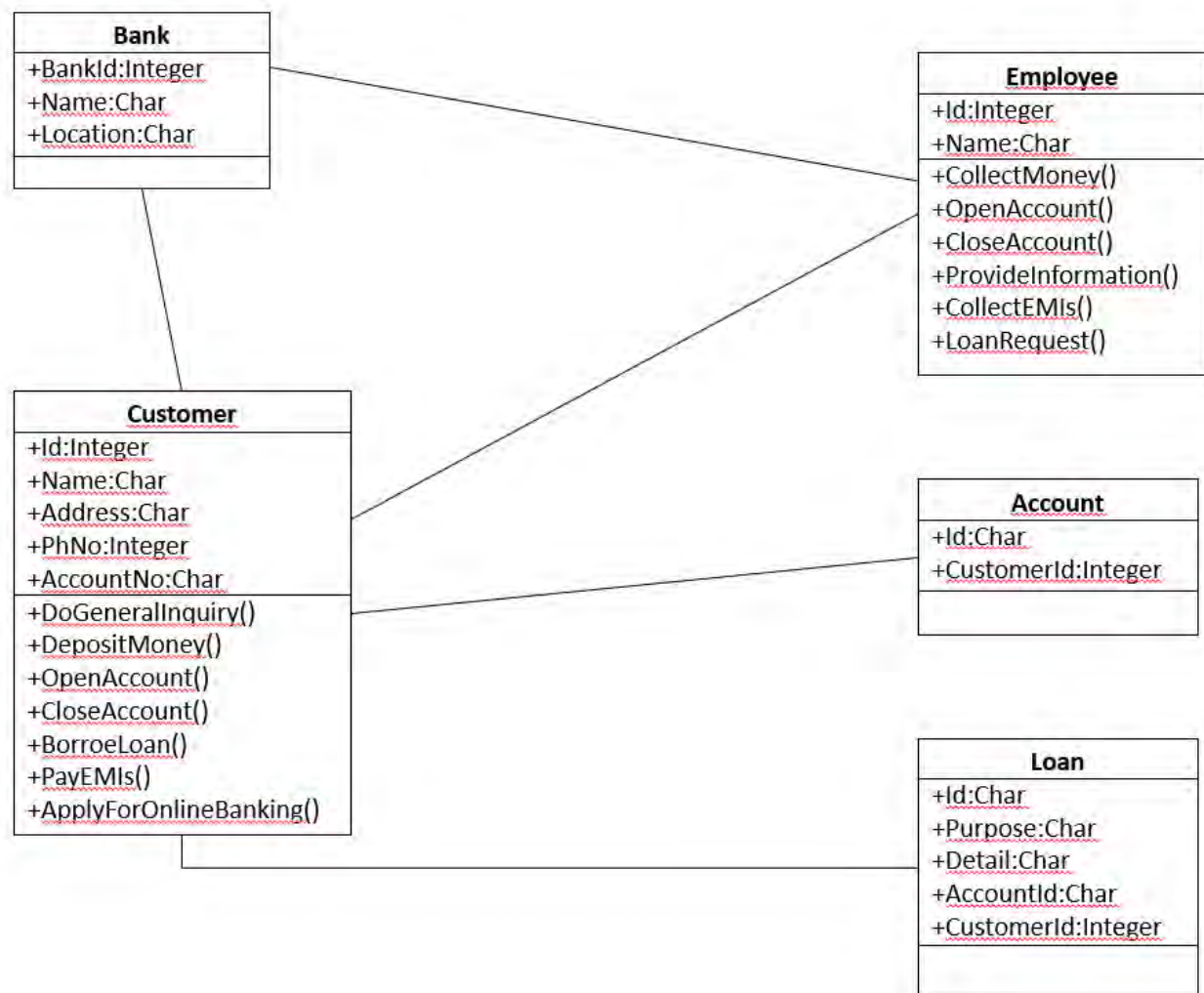


Ilustración 39. Recurso de aprendizaje con grosor de líneas reestructurado

Para concluir, se evalúa nuevamente el recurso de aprendizaje 154Loan.pptx, el resultado se muestra en la Ilustración 40, lo que indica que este recurso de aprendizaje ha sido reestructurado exitosamente y ahora cumple con todos los atributos de calidad obteniendo un nivel de correctitud del 100%.

Weight	Criteria	Check
7%	Slide Background	✓
14%	Line Color	✓
10%	Line Width	✓
8%	Shape width	✓
7%	Shape Background	✓
10%	Font Size	✓
10%	Font Color	✓
34%	Blurring	✓

The score is 100%

Ilustración 40. Nivel de correctitud final del recurso de aprendizaje “154Loan.pptx”

Para este caso de estudio en particular, se desea conocer los valores de los atributos antes y después del proceso de reestructura. En la Tabla 22 se muestran dichos resultados. Los valores de la columna Valor Antes corresponden a la lectura de los valores de los atributos de calidad antes del proceso de reestructura, la columna Valor Después muestra los valores alcanzados después del proceso de reestructura.

Tabla 22. Tabla comparativa de los valores de los atributos

Atributo	Valor	
	Antes	Después
Slide Background	RGB(237,125,49)	RGB(255,255,255)
Line Color	RGB(0,0,0)	RGB(0,0,0)
Line width	2.25	1.0
Shape width	0.5,1.0	1.0
Shape background	RGB(47,85,151)	RGB(255,255,255)
Font size	12	14
Font color	RGB(255,255,255)	RGB(0,0,0)
Blurring	False	False

Dentro del campo Shape width se consideran dos valores debido a que en PowerPoint para representar una clase se considera por un lado el grosor de la forma y por otro también se lee el grosor de los elementos que están incluidos, específicamente las líneas que conforman el clasificador de clases, en consecuencia para cumplir con este atributo todos los elementos deberán tener valores válidos de lo contrario, se considera un incumplimiento. Por otro lado, los resultados de evaluación de las pruebas realizadas antes y después del proceso de reestructura se presentan en la Ilustración

41 en un fragmento de una tabla de Excel. En el Anexo 3 se puede consultar la tabla completa que incluye el resultado de evaluación para cada caso de prueba.

Recurso	Proceso	Font size	Shape width	Line width	Slide backgr	Shape backg	Line color	Font color	Blurring	% de Correct
24order	Antes		✓	✓	✓	✓		✓	✓	76
	Después	✓	✓	✓	✓	✓	✓	✓	✓	100
25window	Antes	✓			✓		✓	✓	✓	75
	Después	✓	✓	✓	✓	✓	✓	✓	✓	100
27Flight	Antes		✓	✓	✓	✓	✓	✓	✓	90
	Después	✓	✓	✓	✓	✓	✓	✓	✓	100
30blast	Antes				✓	✓	✓	✓	✓	72
	Después	✓	✓	✓	✓	✓	✓	✓	✓	100
31atm	Antes		✓	✓	✓			✓		35
	Después	✓	✓	✓	✓	✓	✓	✓		66
32persona	Antes				✓	✓	✓	✓		38
	Después	✓	✓	✓	✓	✓	✓	✓		66
33Escuela	Antes	✓			✓			✓	✓	61
	Después	✓	✓	✓	✓	✓	✓	✓	✓	100
34buscador	Antes				✓			✓		17
	Después	✓	✓	✓	✓	✓	✓	✓		66
35empresa	Antes	✓			✓	✓	✓	✓		48
	Después	✓	✓	✓	✓	✓	✓	✓		66
36hangout	Antes		✓	✓	✓	✓	✓	✓		56
	Después	✓	✓	✓	✓	✓	✓	✓		66
37pedido	Antes	✓	✓	✓	✓	✓	✓	✓	✓	100
	Después	✓	✓	✓	✓	✓	✓	✓	✓	100
38figura	Antes	✓	✓	✓	✓			✓	✓	79
	Después	✓	✓	✓	✓	✓	✓	✓	✓	100
39pedido	Antes				✓	✓		✓		24
	Después	✓	✓	✓	✓	✓	✓	✓		66

Ilustración 41. Resultados de evaluación, antes y después de usar ARE System.

Los resultados de evaluación se presentan mediante gráficas de barras de la Ilustración 42 a la Ilustración 49. Las siguientes gráficas muestran el concentrado de los resultados de evaluación. Para una mejor apreciación de las gráficas, los datos se han organizado en conjuntos más pequeños. De acuerdo con los resultados obtenidos, el porcentaje de correctitud promedio de los recursos de aprendizaje antes del proceso de reestructura es del 64%, de los cuales el 8% cumple en un 100% con los atributos de correctitud, mientras que un 40% representa los recursos de aprendizaje con menos del 60% de correctitud. Después de realizar el proceso de reestructura con la herramienta desarrollada, el 60% alcanzó con éxito el 100% de correctitud, lo que representa un 86% en promedio de correctitud. Los valores mínimos de correctitud antes y después del proceso de reestructura son del 17% y del 66% respectivamente. En el caso de los recursos de aprendizaje que presentan el atributo Nitidez representan el 40%, de los cuales no se logró el 100% de correctitud debido a que la biblioteca Apache POI tiene como limitante el no poder modificar este atributo dentro de las diapositivas.

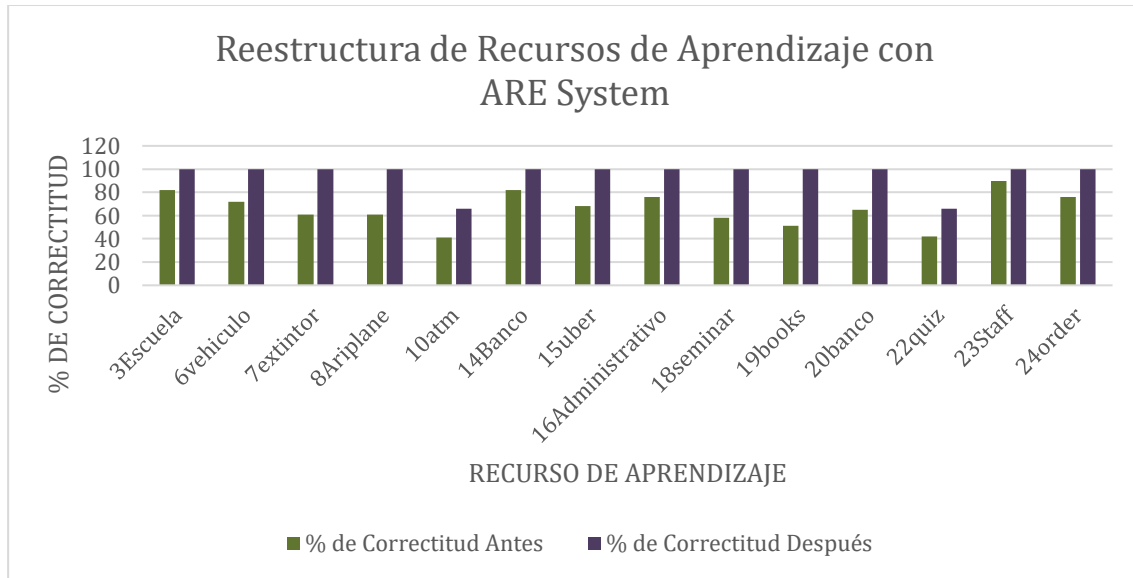


Ilustración 42. Resultados de la evaluación antes y después de usar ARE System

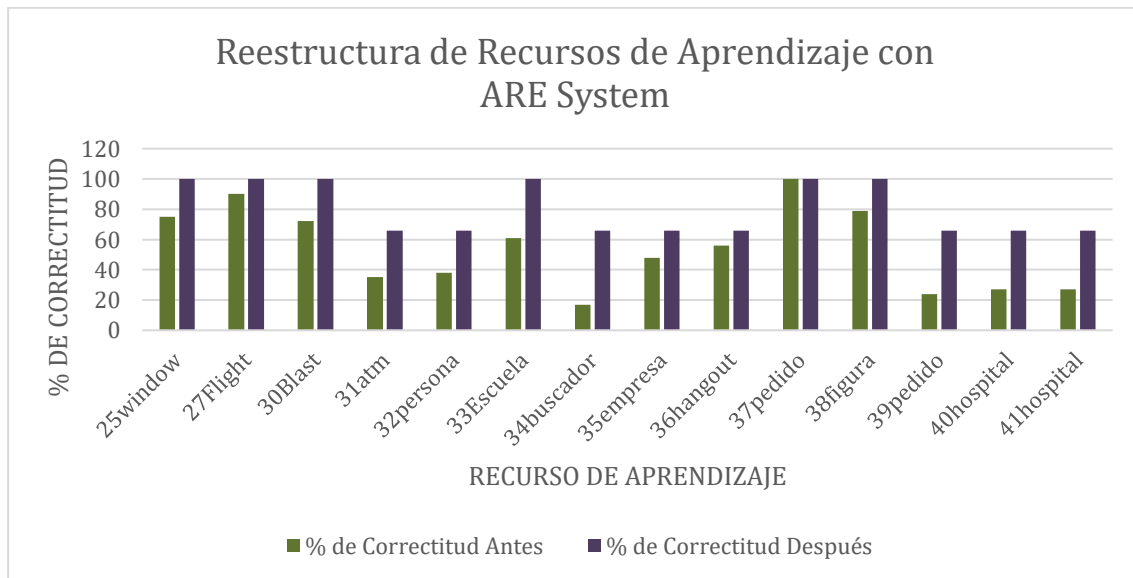


Ilustración 43. Resultados de la evaluación antes y después de usar ARE System

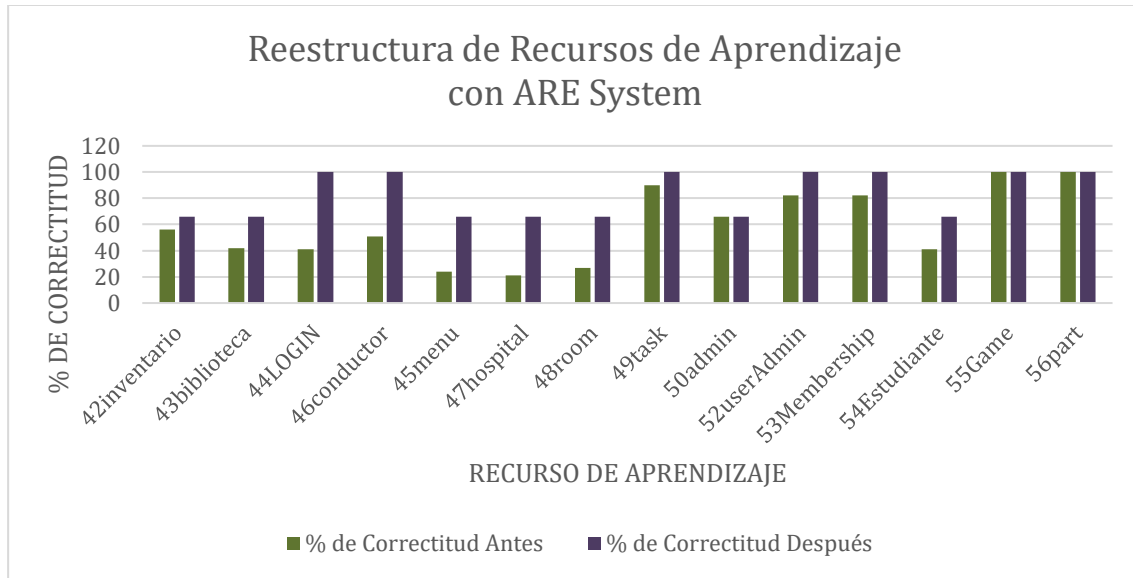


Ilustración 44. Resultados de la evaluación antes y después de usar ARE System

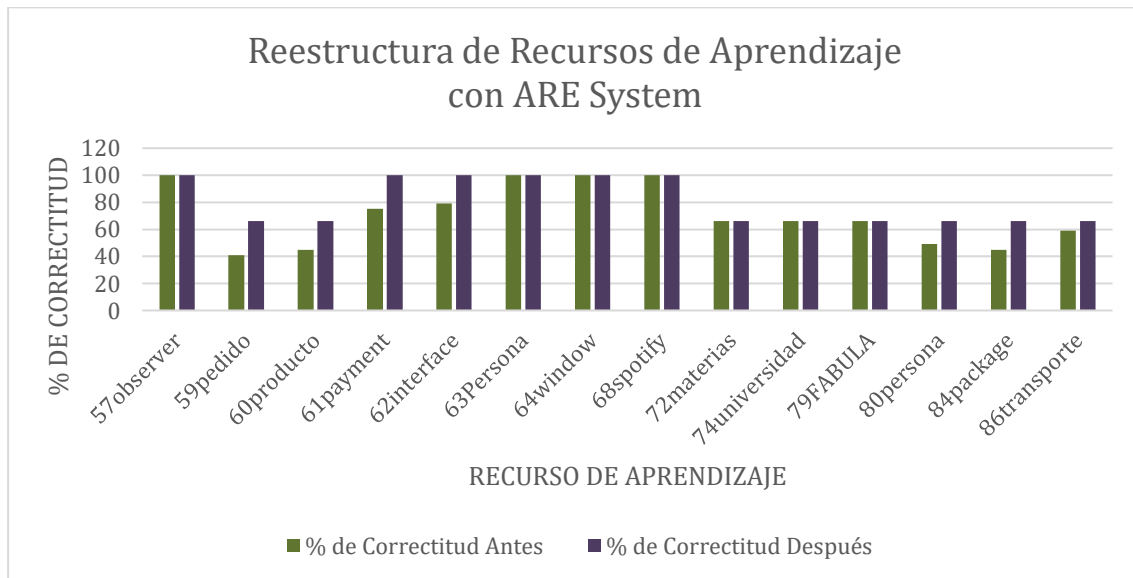


Ilustración 45. Resultados de la evaluación antes y después de usar ARE System

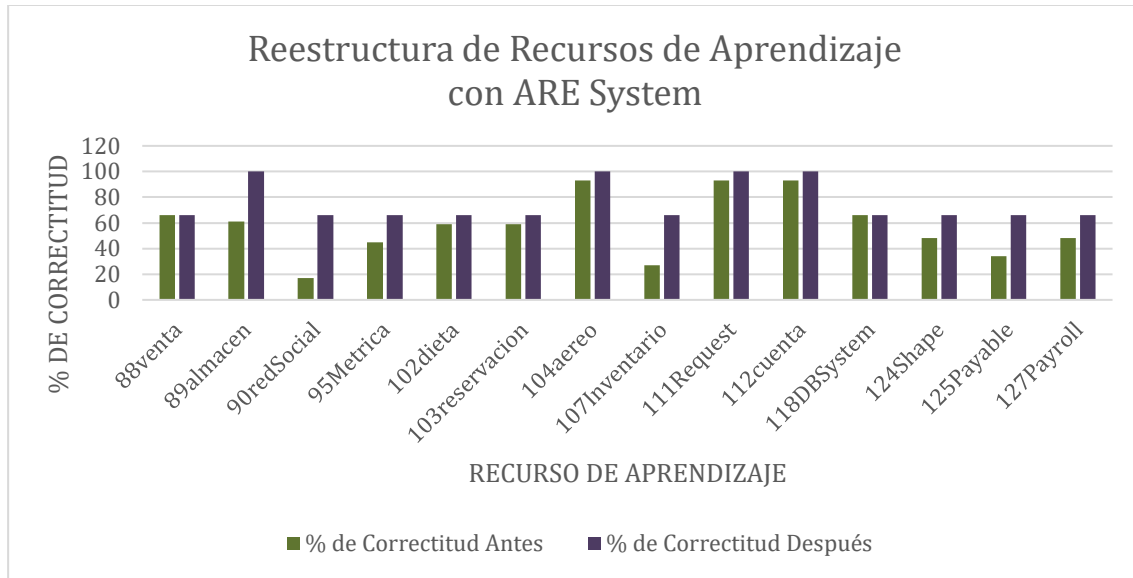


Ilustración 46. Resultados de la evaluación antes y después de usar ARE System

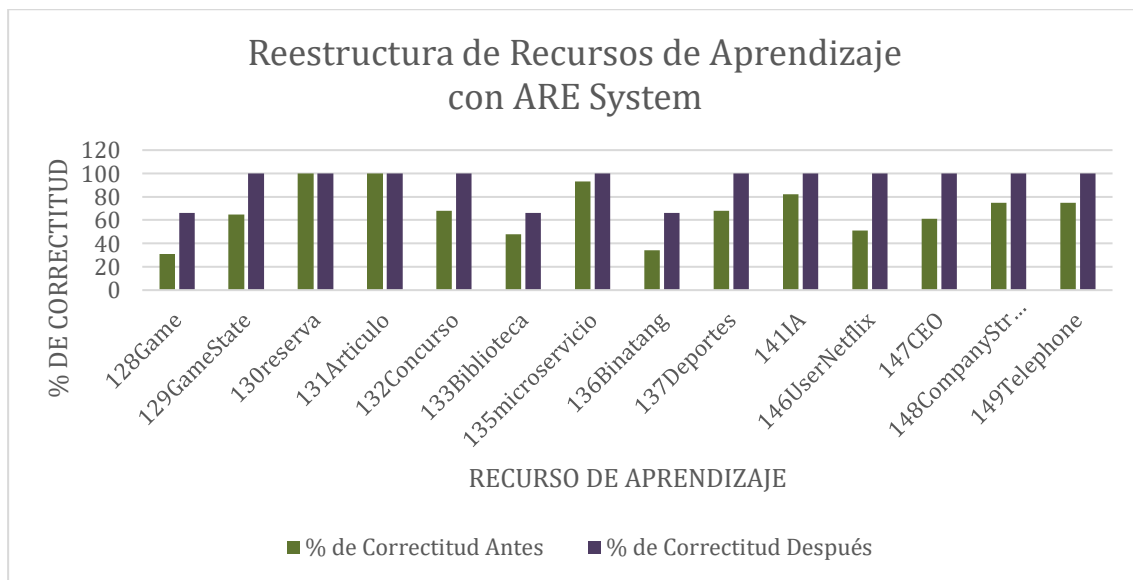


Ilustración 47. Resultados de la evaluación antes y después de usar ARE System

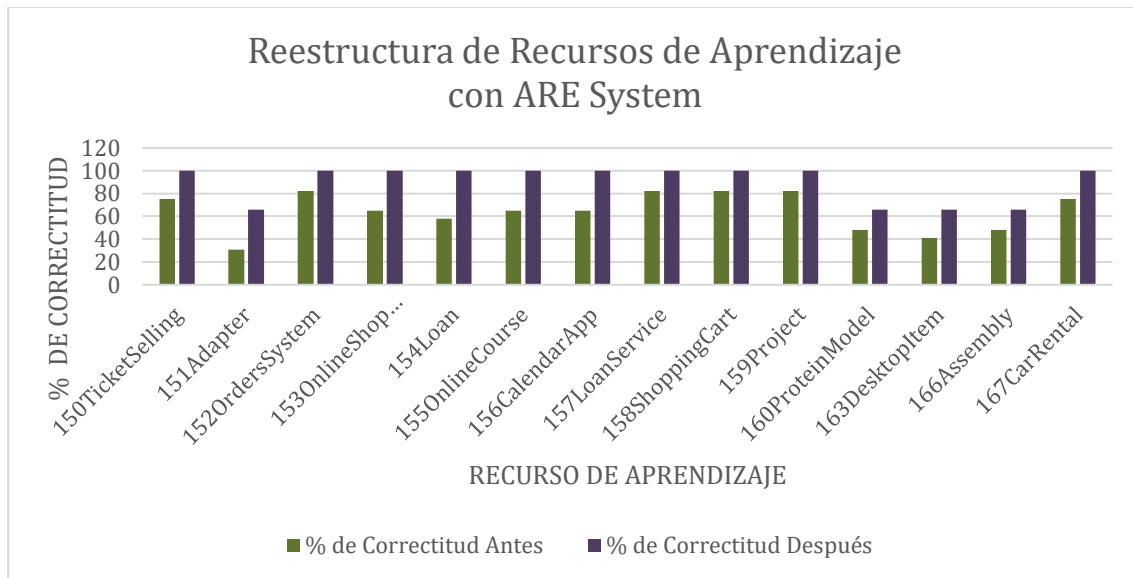


Ilustración 48. Resultados de la evaluación antes y después de usar ARE System

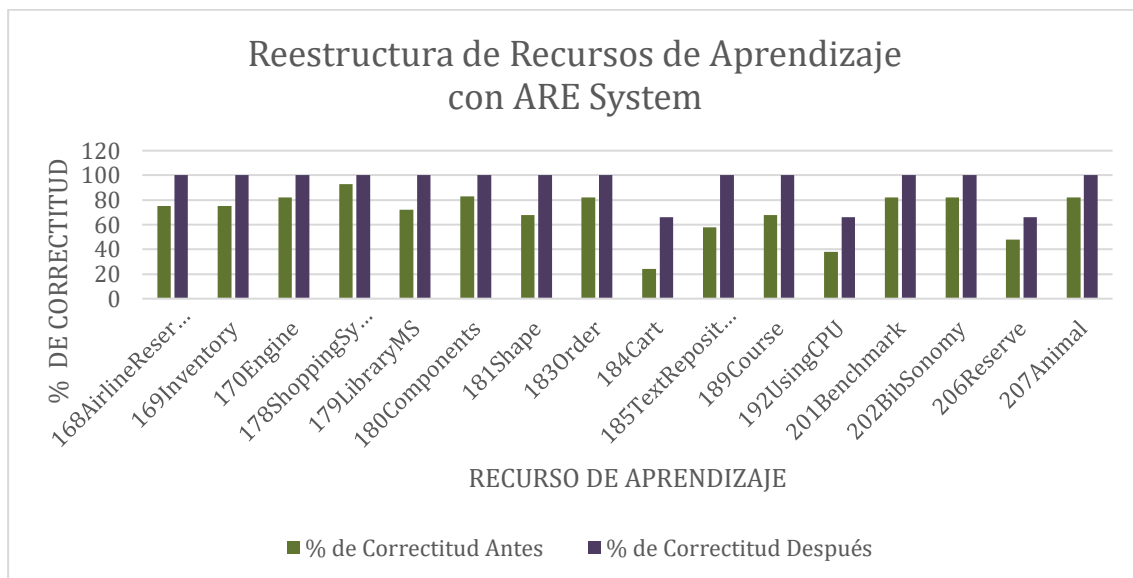


Ilustración 49. Resultados de la evaluación antes y después de usar ARE System

En un análisis preliminar, se encontró que ningún recurso de aprendizaje cumplía el tamaño de fuente, sin embargo, esos resultados fueron descartados dado que el tamaño de texto sugerido se aplica en documentos. Para efectos de este estudio se recomienda el tamaño de fuente de 14 pixeles. Los resultados se resumen en la Tabla 23 y se representan mediante una gráfica de barras en la Ilustración 50. Como parte del análisis se observó que el atributo con más incidencias fue el grosor de líneas, seguido del color de fondo en las clases. Por el contrario, el color de fondo en el panel de diapositivas es el atributo que se presentó menos veces.

Tabla 23. Número de incidencias

Atributo de calidad	No. de incidencias
Slide Background	1
Shape width	74
Line width	74
Font Size	13
Shape Background	52
Line Color	37
Font Color	13
Blurring	45

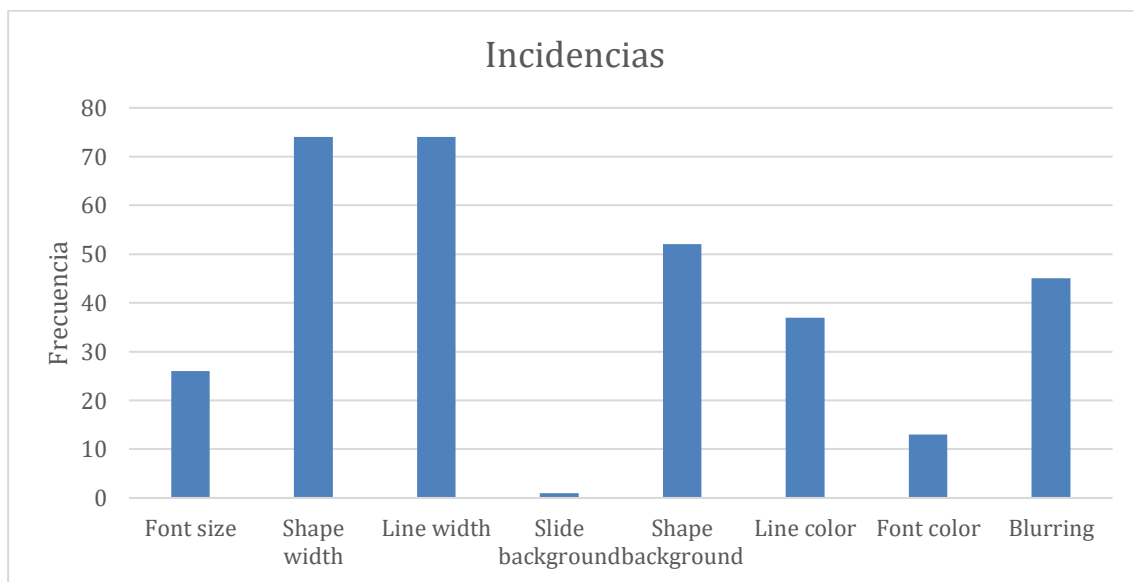


Ilustración 50. Número de incidencias de defectos de los diagramas de clases

En la Ilustración 51 también se incluye la matriz de correlación de los diferentes atributos de calidad, en la cual los valores varían entre -1 y +1, donde un valor cercano a +1 indica una correlación positiva fuerte y un valor -1 indica una correlación negativa fuerte, si el valor es cercano a 0 se interpreta que no existe una correlación entre las variables. En la matriz se observa que Shape Width y Line Width tienen una correlación fuerte, lo que implica que el espesor de las figuras es consistente dentro de los diagramas. Otra correlación es la del atributo Font Size con Line Color, lo que puede indicar que el color de las relaciones está en función del texto o viceversa, lo cual se puede entender que, si cambia uno el otro también, bajo el supuesto de que se busca un contraste entre ambos atributos. Otro aspecto más se presenta con Font Color contra Shape Background y Slide Background, lo cual se puede ver cuando las formas tienen un color distinto a blanco y lo que se se busca en los diagramas, es contrastar de manera natural entre el fondo del área de trabajo y las letras del diagrama.

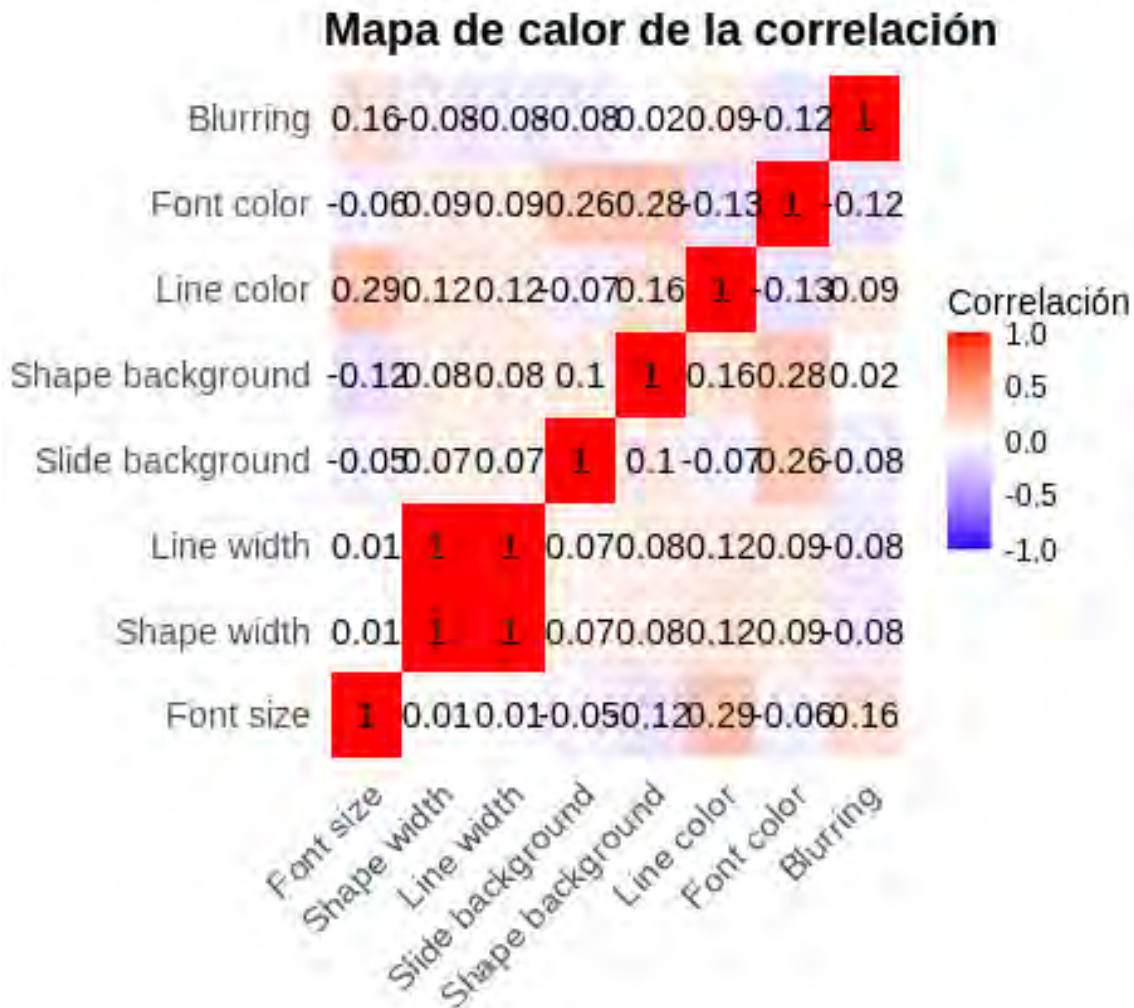


Ilustración 51. Mapa de correlación entre los diferentes atributos de calidad

Estas pruebas revelan que el sistema hace los cambios correctamente, mostrando que el sistema es capaz de realizar el proceso de reestructura con éxito, logrando con ello el cumplimiento de los objetivos planteados en este documento.

V. Conclusiones

Con la implementación de esta herramienta se muestra que es posible reestructurar elementos que ayudan a mejorar aspectos relacionados al dominio de legibilidad de diagramas de clases UML usados como recursos de aprendizaje. El uso de la biblioteca Apache POI en el desarrollo de este sistema fue un acierto que permitió realizar el proceso de reestructura de los recursos de aprendizaje. Esta biblioteca es la base fundamental en el proceso de desarrollo de “ARE System”, sin embargo, la limitante encontrada hasta ahora es que solo es posible evaluar el atributo nitidez. Los recursos de aprendizaje que presentan esta característica al ser un atributo que no se

puede reestructurar con Apache POI, logran un 66% de correctitud, en comparación al 100% que obtienen el resto de los recursos de aprendizaje.

Como parte del desarrollo de la primera fase de la herramienta, se cuenta con los módulos para Registrar usuarios, Evaluar recurso de aprendizaje, Procesar recurso de aprendizaje y Descargar recurso de aprendizaje. El módulo Registrar usuario permite dar de alta a usuarios para fines estadísticos, el módulo Evaluar ayuda a determinar la calidad de un recurso de aprendizaje; Procesar recurso realiza los cambios necesarios para reestructurar o modificar los atributos de calidad de los recursos de aprendizaje; finalmente, el módulo Descargar recurso pone a disposición del usuario los recursos de aprendizaje reestructurados. Con la implementación de ARESystem, se determina que es posible modificar aspectos asociados con la legibilidad de elementos como son el tamaño y color del texto, el color y grosor de las líneas, así como de modificar los colores, todo ello con el fin de mejorar la calidad visual del material educativo. Los lineamientos de reestructura y los valores de los atributos se basan en el Modelo de Correctitud (Sandoval, 2023). De este modo se comprueba la viabilidad para modificar aspectos relacionados con la legibilidad de diagramas de clases UML, es decir, mejorar aspectos asociados con la parte estética del recurso de aprendizaje. En cuanto al tratamiento del texto, en el Modelo de Correctitud se sugiere un tamaño de fuente de 12 pixeles, sin embargo, esta medida es aconsejable aplicarlo en documentos, por lo que en este trabajo se sugiere un tamaño de fuente de 14 pixeles. Estos cambios ayudan a mejorar la apariencia dentro de los clasificadores de clase, evitando con ello el traslape del texto, mejorando de este modo el esparcimiento y dándole uniformidad al texto contenido dentro de los objetos que representan las clases en PowerPoint.

Además, con el desarrollo de esta herramienta, se permite automatizar el proceso de reestructura así como el proceso de evaluación de los diagramas de clases UML usados como recursos de aprendizaje, disminuyendo con ello el tiempo invertido en la realización de estas tareas. Por otro lado, es posible personalizar un recurso de aprendizaje permitiendo seleccionar de una lista de atributos aquellos que se desean reestructurar, dando la ventaja y la libertad de mejorar el recurso de aprendizaje según las necesidades de cada usuario. De manera general, el desarrollo de esta herramienta contribuye a que quienes consumen estos recursos logren una mayor concentración en el contenido esencial, evitando distracciones por elementos gráficos innecesarios.

En la parte de evaluación, el sistema determina el nivel de correctitud del recurso de aprendizaje asignando una ponderación que varía entre 0 y 100, donde 100 representa un recurso de aprendizaje que cumple con todos los atributos de calidad (calificación más alta) y 0 para los diagramas de clases que no cumplen o no cuentan con las especificaciones de diseño (calificación más baja). La correctitud del recurso está en

función de los atributos que si se cumplen, esto debido a que no todos los atributos tienen la misma puntuación, indicando con ello el nivel de correctitud alcanzado para cada recurso de aprendizaje. Sin embargo, con base en los resultados obtenidos, se concluye que ARE System ayuda a mejorar la parte estética del recurso de aprendizaje, permitiendo transformar los atributos de legibilidad tales como el tamaño del texto, el color de los objetos que representan las clases, el color de las líneas que representan las relaciones entre clases, así como del texto, logrando con ello una mejor apreciación en cuestiones de visibilidad.

Como parte de los trabajos a futuro, se incluye la capacidad de evaluar la correcta implementación de clases, herencias y relaciones dentro del paradigma Orientado a Objetos, se observa que para lograr lo anterior, es necesario identificar la segmentación de objetos para establecer las formas, tamaños y posiciones de los objetos que conforman los diagramas de clases. Además de incorporar la implementación de los atributos asociados con la legibilidad de líneas, así como la integración del módulo de Correctitud de Dominio UML. Por lo tanto, al haber realizado las pruebas necesarias con ARESystem y de acuerdo con los resultados obtenidos, se concluye que el sistema reestructura los elementos asociados a un diagrama de clases y que el Modelo de Correctitud usado como referencia cumple con los objetivos establecidos en el apartado de Legibilidad. Con estas mejoras se ofrece al usuario un recurso de aprendizaje fácil de leer.

VI. Referencias

- 1 Aprendizaje virtual. (2014). México: Editorial Digital UNID.
- 2 Vásquez, M., & Alexandra, J. (2015). “Los recursos multimedia y el aprendizaje de la prematemática de los estudiantes del centro de desarrollo infantil Celitas Cantón Ambato, provincia de Tungurahua”.
- 3 Coscollá, P. (2009). Material de clase: presentaciones.
- 4 Educativa, D.D. (2015). Material de aprendizaje: material de clase.
- 5 Universitat Oberta de Catalunya (UOC). (2015). Guía de herramientas y recursos. eLearn Center. https://www.uoc.edu/portal/_resources/ES/documents/elearncenter/guias_modelo_educativo/Guia_herramientas_y_recursos_ES.pdf
- 6 Universidad Nacional Autónoma de México (UNAM). (2024). Glosario. Recrea CUAIEED. <https://recrea.cuaieed.unam.mx/glosario/>
- 7 W3Schools. (s.f.). XML - What is XML? W3Schools. https://www.w3schools.com/xml/xml_what.asp
- 8 Object Management Group. (s.f.). About UML 2.5.1. OMG. <https://www.omg.org/spec/UML/2.5.1/About-UML/>
- 9 Fenton, N. E., Pfleeger, S. L. (1997). Software Metrics: A Rigorous and Practical Approach. Alemania: International Thomson Computer Press.
- 10 Rumbaugh, J. (2007). El lenguaje unificado de modelado 2/e. España: Pearson Educación. (verificar)
- 11 Debrauwer, L., Van der Heyde, F. (2016). UML 2.5: iniciación, ejemplos y ejercicios corregidos. España: Ediciones ENI.
- 12 Blasco, F. (2019). Programación Orientada a Objetos en JAVA. España: RA-MA S.A. Editorial y Publicaciones.
- 13 Kimmel, P. (2011). MANUAL DE UML. España: McGraw-Hill Interamericana de España S.L.
- 14 Rumbaugh, J., Booch, G., Jacobson, I. (2004). El lenguaje unificado de modelado.

Manual de referencia. España: Addison-Wesley.

- 15 Teniente López, E., Olivé Ramon, A., Mayol Sarroca, E., Gómez Seone, C. (2004). Diseño de sistemas software en UML. España: Universitat Politecnica de Catalunya. Iniciativa Digital Politécnica.
- 16 Piattini Velthuis, M. G., García Rubio, F. Ó. (2019). Medición de Software. España: RA-MA S.A. Editorial y Publicaciones.
- 17 REAL ACADEMIA ESPAÑOLA: Diccionario de la lengua española, 23.^a ed., [versión 23.8 en línea]. <<https://dle.rae.es>> [25/05/25].
- 18 UNED. (2008). Glosario. Cátedra UNESCO de Educación a Distancia. https://campus.eco.unlpam.edu.ar/pluginfile.php/9611/mod_resource/content/3/9-Glosario_EAD.pdf
- 19 "ISO/IEC/IEEE International Standard - Systems and software engineering -- Vocabulary," in ISO/IEC/IEEE 24765:2010(E), vol., no., pp.1-418, 15 Dec. 2010,
- 20 IEEE Standard Glossary of Software Engineering Terminology. (n.d.).
- 21 ISO 25000. (s.f.). ISO 25010. ISO 25000. <https://iso25000.com/index.php/normas-iso-25000/iso-25010>
- 22 Garrido-Hernansaiz, Helena & Rodríguez-Rey, Rocío. (2017). Evaluación y Mejora de la Calidad de la Presentación de Diapositivas como Herramienta Docente Universitaria.
- 23 Flores-Hernández, Fernando, & Villamar-Chulín, Joel. (2012). Elementos para el desarrollo de presentaciones académicas en PowerPoint. Investigación en educación médica, 1(3), 152-156.
- 24 Alarcón, G. J. M., Díaz, J. M. M., Hernández, M. A. S., Niño, U. A. F. y Leal, H. V. (2024). Aprendizaje de Word y PowerPoint en Estudiantes de Secundaria. Anuario de Psicología Educativa Aplicada, 1, Artículo e4.
- 25 Rajngewerc, M. Loto, P.A. (2022) Buenas prácticas para crear presentaciones accesibles. Recuperado de: <https://www.metadocencia.org/post/presentaciones-accesibles/> el 17/05/2025.
- 26 Dabo Sun, & Wong, K. (2005). On Evaluating the Layout of UML Class Diagrams for Program Comprehension. 13th International Workshop on Program Comprehension (IWPC'05). doi:10.1109/wpc.2005.26
- 27 Napa Vilela, Z. A., (2023). Los recursos didácticos como apoyo en el proceso de enseñanza-aprendizaje de los estudiantes. MQRInvestigar, 7(3), 4078–4105.

<https://doi.org/10.56048/MQR20225.7.3.2023.4078-4105>

- 28 Pinto, M., Gomez-Camarero, C., & Fernández-Ramos, A. (2012). Los recursos educativos electrónicos: perspectivas y herramientas de evaluación. *Perspectivas Em Ciência Da Informação*, 17(3), 82–99. doi:10.1590/s1413-99362012000300007
- 29 Oliveira, R. D. de. (2020). Análise do uso da cor no Diagrama de Classes da Linguagem Unificada de Modelagem (UML) | Analysis of the use of color in the Unified Modeling Language Class Diagram (UML). *InfoDesign*, 17(1), 116–130. <https://doi.org/10.51358/id.v17i1.783>
- 30 Hurtado Huicochea, Antonio. (2023). Descripción de criterios sobre presentaciones digitales en Power Point con estudiantes del curso postuniversitario en competencias docentes. *RIDE. Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, 13(26), e042. Epub 08 de septiembre de 2023. <https://doi.org/10.23913/ride.v13i26.1481>
- 31 Oyama, S., & Yamasaki, T. (2019). Visual Clarity Analysis and Improvement Support for Presentation Slides. 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM), 421-428.
- 32 Holakopour, L., Azizifar, A., & Gowhari, H. (2014). The Effect of Power Point on Reading Comprehension Improvement among High school students: A case Study in the City of Shoush.
- 33 Borade, J.G., Kiwelekar, A.W., Netak, L.D. (2022). Automated grading of PowerPoint presentations using latent semantic analysis. *Revue d'Intelligence Artificielle*, Vol. 36, No. 2, pp. 305-311.
- 34 López González, V., Palmeros y Ávila, G., & Coeto Calcáneo, I. A. (2021). Guía para la evaluación y adecuación de materiales digitales accesibles. *Emerging Trends in Education*, 3(5). <https://doi.org/10.19136/etie.a3n5.3796> (Original work published 2020)
- 35 J. Sandoval Acosta, Uso de reglas de dominio para evaluar recursos de servicios web de aprendizaje. Cuernavaca: Cenidet 2023
- 36 O. G. Fragoso-Díaz, J. A. Sandoval-Acosta, F. J. Álvarez-Rodríguez, J. C. Rojas-Pérez and R. Santaolaya-Salgado, "Systematic Review of Quality in Class Diagrams for Software Engineering Competencies," in *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, vol. 17, no. 4, pp. 351-357, nov. 2022
- 37 Castillo Badillo, E. F., Fragoso-Díaz, O. G., Rojas Pérez, J. C., & Valenzuela Robles, B. D. (2023). QLERR (Quality Learning Resources Reader): Un sistema para la evaluación de recursos de aprendizaje. *Tecnología y Ciencia Aplicada*,

6(1), 51–52. <https://jcyta.cenidet.tecnm.mx/>

- 38 F. Pérez Rodríguez, Marco de servicios para medición de la calidad de formato de Recursos de Aprendizaje en Servicios Web. Cuernavaca: Cenidet 2017
- 39 J. Escobar Megchún, Generador de Servicios Web de Aprendizaje Compuestos a partir de Recursos Educativos. Cuernavaca: Cenidet 2017
- 40 H. Salinas Román, Marco Orientado a Objetos para Medir la Calidad en Servicios Web de Aprendizaje. Cuernavaca: Cenidet 2017
- 41 L Rebolledo Castañeda, Análisis y Definición de una Métrica para Evaluar la Legibilidad en Servicios Web de Aprendizaje. Cuernavaca: Cenidet 2017
- 42 Thakur, J. S., & Gupta, A. (2014). Automatic generation of sequence diagram from use case specification. Proceedings of the 7th India Software Engineering Conference on - ISEC '14.
- 43 Thakur, J. S., & Gupta, A. (2016). AnModeler: a tool for generating domain models from textual specifications. Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering - ASE 2016.
- 44 Eichelberger, H., & Gudenberg, J.W. (2003). UML Class Diagrams - State of the Art in Layout Techniques. *IEEE Working Conference on Software Visualization*.
- 45 Gudenberg, J. W. v., Niederle, A., Ebner, M., & Eichelberger, H. (2006). Evolutionary layout of UML class diagrams. Proceedings of the 2006 ACM Symposium on Software Visualization - SoftVis '06.
- 46 B. Karasneh, R. Jolak and M. R. V. Chaudron, "Using Examples for Teaching Software Design: An Experiment Using a Repository of UML Class Diagrams," 2015 Asia-Pacific Software Engineering Conference (APSEC), New Delhi, India, 2015, pp. 261-268,
- 47 S. Freire, A. Passos, M. Mendonça, C. Sant'Anna and R. O. Spínola, "On the Influence of UML Class Diagrams Refactoring on Code Debt: A Family of Replicated Empirical Studies," 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Portoroz, Slovenia, 2020, pp. 346-353
- 48 P. Pradhan, A. K. Dwivedi and S. K. Rath, "Impact of Design Patterns on Quantitative Assessment of Quality Parameters," 2015 Second International Conference on Advances in Computing and Communication Engineering, Dehradun, India, 2015, pp. 577-582
- 49 Faitelson, D., & Tyszberowicz, S. (2017). UML Diagram Refinement (Focusing on

Class-and Use Case Diagrams). 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE).

- 50 Ben Abdesslem Karaa, W., Ben Azzouz, Z., Singh, A., Dey, N., S. Ashour, A., & Ben Ghazala, H. (2015). Automatic builder of class diagram (ABCD): an application of UML generation from functional requirements. *Software: Practice and Experience*, 46(11), 1443–1458.
- 51 Baqais, A. & Alshayeb, M. (2016). Automatic Refactoring of Single and Multiple-view UML Models using Artificial Intelligence Algorithms. Department of Information Science and Technology, King Fahd University of Petroleum & Minerals, KFUPM Blvd., Dhahran, Saudi Arabia
- 52 Ali, N. H., Shukur, Z., & Idris, S. (2007). A Design of an Assessment System for UML Class Diagram. 2007 International Conference on Computational Science and Its Applications (ICCSA 2007).
- 53 Z. Liu, X. Li, J. Liu, and J. He, “Integrating and refining UML models,” in *Workshop on Consistency Problems in UML-Based Software Development: Understanding and Usage of Dependency Relationships*, 2004, pp. 23–40.
- 54 Wagner, G. (2005). Rule Modeling and Markup. In: Eisinger, N., Małuszyński, J. (eds) *Reasoning Web. Lecture Notes in Computer Science*, vol 3564. Springer, Berlin, Heidelberg.
- 55 Rajesh, S., & Chandrasekar, A. (2016). An Efficient Object-Oriented Design Model: By Measuring and Prioritizing the Design Metrics of UML Class Diagram with Preeminent Quality Attributes. *Indian Journal of Science and Technology*, 9(21).
- 56 Bian, W., Alam, O., & Kienzle, J. (2019). Automated Grading of Class Diagrams. 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C).
- 57 Al-Khiaty, M. A.-R., & Ahmed, M. (2014). Similarity assessment of UML class diagrams using simulated annealing. 2014 IEEE 5th International Conference on Software Engineering and Service Science.
- 58 Doan, K.-H., & Gogolla, M. (2019). Quality Improvement for UML and OCL Models Through Bad Smell and Metrics Definition. 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C).
- 59 Sheng, F., Zhu, H., & Yang, Z. (2019). Towards the Mechanized Semantics and Refinement of UML Class Diagrams. 2019 26th Asia-Pacific Software Engineering Conference (APSEC).

- 60 Streckenbach, M., & Snelting, G. (2004). Refactoring class hierarchies with KABA. *ACM SIGPLAN Notices*, 39(10), 315.
- 61 Shaikh, A., & Wiil, U. K. (2018). Overview of Slicing and Feedback Techniques for Efficient Verification of UML/OCL Class Diagrams. *IEEE Access*, 6, 23864–23882.
- 62 Hafeez Khan, A., Hyder Abbas Musavi, S., Rehman, A.-U., & Shaikh, A. (2018). Ontology-Based Finite Satisfiability of UML Class Model. *IEEE Access*, 6, 3040–3050.
- 63 Dinh-Trong, T., Kawane, N., Ghosh, S., France, R., & Andrews, A. A. (n.d.). A Tool-Supported Approach to Testing UML Design Models. 10th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'05).
- 64 Störrle, H. (2014). On the Impact of Layout Quality to Understanding UML Diagrams: Size Matters. *Model-Driven Engineering Languages and Systems*, 518–534.
- 65 Sidhu, B. K., Singh, K., & Sharma, N. (2018). A Catalogue of Model Smells and Refactoring Operations for Object-Oriented Software. 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT).
- 66 Nikiforova, O., & Pavlova, N. (2008). Development of the Tool for Generation of UML Class Diagram from Two-Hemisphere Model. 2008 The Third International Conference on Software Engineering Advances.
- 67 Gutwenger, C., Jünger, M., Klein, K., Kupke, J., Leipert, S., & Mutzel, P. (2003). A new approach for visualizing UML class diagrams. *Proceedings of the 2003 ACM Symposium on Software Visualization - SoftVis '03*.
- 68 Barbosa, L. S., & Meng, S. (2008). UML Model Refactoring as Refinement: A Coalgebraic Perspective. 2008 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing.
- 69 Maddeh, M., Ayouni, S., Alyahya, S., & Hajjej, F. (2021). Decision tree-based Design Defects Detection. *IEEE Access*, 9, 71606–71614.
- 70 Caballé, S., & Cabot, J., & Clarisó, R., & Conesa, J., & Planas, E., & Riera Terrén, D. (2009). CUCKOO: Una plataforma web para la verificación de modelos UML. *Asociación de Enseñantes Universitarios de la Informática (AENUI)*
- 71 Ho-Quang, T., Chaudron, M. R. V., Samuelsson, I., Hjaltason, J., Karasneh, B., & Osman, H. (2014). Automatic Classification of UML Class Diagrams from Images. 2014 21st Asia-Pacific Software Engineering Conference.

- 72 Deryugina, O.P., & Nikulchev, E.V. (2021). Methods, Algorithms and Software for the UML Class Diagram Refactoring.
- 73 Marković, S., & Baar, T. (2007). Refactoring OCL annotated UML class diagrams. *Software & Systems Modeling*, 7(1), 25–47.
- 74 Lano, K., & Rahimi, S.K. (2013). Case study: Class diagram restructuring. *Transformation Tool Contest*.
- 75 Massoni, T., Gheyi, R., & Borba, P. (2005). Formal Refactoring for UML Class Diagrams. In *Anais do XIX Simpósio Brasileiro de Engenharia de Software*, (pp. 152-167). Porto Alegre: SBC.
- 76 Hosseini, S., & Azgomi, M.A. (2008). UML Model Refactoring with Emphasis on Behavior Preservation. 2008 2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering, 125-128.
- 77 Dobrzanski, L., & Kuzniarz, L. (2006). An approach to refactoring of executable UML models. *Proceedings of the 2006 ACM Symposium on Applied Computing - SAC '06*.
- 78 Sunyé, G., Pollet, D., Le Traon, Y., & Jézéquel, J.-M. (2001). Refactoring UML Models. *Lecture Notes in Computer Science*, 134–148.
- 79 Van Gorp, P., Stenten, H., Mens, T., & Demeyer, S. (2003). Towards Automating Source-Consistent UML Refactorings. «UML» 2003 - The Unified Modeling Language. *Modeling Languages and Applications*, 144–158.
- 80 Gustav Bergström, Fadhl Hujainah, Truong Ho-Quang, Rodi Jolak, Satrio Adi Rukmono, Arif Nurwidyantoro, Michel R.V. Chaudron, (2022) Evaluating the layout quality of UML class diagrams using machine learning, *Journal of Systems and Software*, Volume 192,2022,111413, ISSN 0164-1212
- 81 Eichelberger, H., & Schmid, K. (2009). Guidelines on the aesthetic quality of UML class diagrams. *Information and Software Technology*, 51(12), 1686–1698.
- 82 Najd Altoyan and Don Batory. (2023). On Proving the Correctness of Refactoring Class Diagrams of MDE Metamodels. *ACM Trans. Softw. Eng. Methodol.* 32, 2, Article 44 (March 2023), 42 pages.
- 83 Chren, Stanislav & Buhnova, Barbora & Macak, Martin & Daubner, Lukáš & Rossi, Bruno. (2019). Mistakes in UML Diagrams: Analysis of Student Projects in a Software Engineering Course
- 84 Enckevort, T. van. (2009). *Refactoring UML models. Proceeding of the 24th ACM*

SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications - OOPSLA '09.

- 85 Boulanger, T. (2015). XML práctico: Bases esenciales, conceptos y casos prácticos (2ª edición). España: Ediciones ENI.
- 86 Parul, & Sidhu, B.K. (2016). Model Smells In Uml Class Diagrams.
- 87 Eichelberger, Holger. (2003). Aesthetics of Class Diagrams (PDF). Visualizing Software for Understanding and Analysis, International Workshop on. 23.
- 88 Institute of Electrical and Electronics Engineers. (1997). *IEEE standard for conceptual modeling language - Syntax and semantics for IDEF1X97 (IDEFobject) (IEEE Std 1320.2-1998)*. IEEE.
- 89 Object Management Group. (s.f.). About UML 2.5.1. OMG. <https://www.omg.org/spec/UML/2.5.1/About-UML>
- 90 Object Management Group: Superstructure, 2005

Anexo 1. Guía de usuario

Contenido

1. Introducción
2. Objetivo de la aplicación
3. Ingresar a la aplicación
4. Registro de nuevos usuarios
5. Importar recurso de aprendizaje
6. Evaluar recurso de aprendizaje
7. Informe de resultados
8. Procesar recurso de aprendizaje
9. Exportar recurso de aprendizaje
10. Salir del sistema

1. Introducción

Este documento está diseñado para ofrecer a los usuarios una visión general sobre la funcionalidad de la aplicación, proporcionando una guía clara y detallada sobre las instrucciones que deben seguirse para garantizar su correcto funcionamiento. Además, la guía busca apoyar a los usuarios a resolver posibles dudas sobre el funcionamiento de la aplicación. El propósito de esta aplicación es analizar, reestructurar y presentar diagramas de clases contenidos en recursos de aprendizaje.

2. Objetivo de la aplicación

Permitir la reestructura de diagramas de clases UML contenidos en recursos de aprendizaje en formato PowerPoint tomados de internet. Las funcionalidades principales de la herramienta son: Iniciar sesión de usuario, Cargar archivo al sistema, Evaluar recurso de aprendizaje, Procesar o reestructurar el recurso de aprendizaje y Descargar el archivo.

3. Ingresar al sistema

En la Ilustración 52 se presenta la página de Bienvenida del sistema. Se deberá pulsar el botón Continuar para ingresar a la página de Inicio.



Ilustración 52. Página “Bienvenida”

Al dar clic en el comando ‘Continuar’ se presenta la página de selección de tipo de ingreso con las opciones Log In o Sign Up. Si usted ya se ha registrado previamente, basta con hacer clic en la opción Log In e introducir el Identificador de usuario y la contraseña, el formato de la pantalla de ingreso se muestra en la Ilustración 53; en caso contrario, vaya al paso 4.

Tecnológico Nacional de México (TecNM)

Centro Nacional de Investigación y Desarrollo Tecnológico (Cenidet)

Automated Restructuring System

Log In

Sign Up

Login

User:

aredes@correo.com

Password:

.....

Send data

Clean data

Ilustración 53. Página Log In

Elementos del botón Log In

- User: Deberá ingresar su correo electrónico.
- Password: Deberá ingresar su contraseña.
- Send data: Haga clic en este botón para confirmar la operación de ingreso.
- Clean data: Haga clic en este botón para borrar la información capturada.

4. Registro de nuevos usuarios

Si es la primera vez que usted ingresa al sistema, será necesario capturar la siguiente información para darse de alta. Posteriormente podrá iniciar sesión, una vez completado el formulario de registro mostrado en la Ilustración 54.

- CURP: Ingrese en este campo su clave de registro de población.
- Name: Ingrese en este campo su nombre.
- Last name: Ingrese en este campo su apellido.
- Password: Ingrese en este campo la clave de acceso al sistema.

- Confirm password: Ingrese en este campo la clave de acceso antes capturada para la confirmación de la contraseña. Esta información deberá guardarla y conservarla en un lugar seguro.
- User: El sistema le asignará un identificador de usuario con el cual podrá ingresar al sistema. Esta información deberá guardarla y conservarla en un lugar seguro.
- Send data: Haga clic en este botón para confirmar su registro en el sistema.
- Clean data: Permite borrar todo el contenido del formulario, una vez que se han eliminado los datos no es posible su recuperación.

Tecnológico Nacional de México (TecNM)

Centro Nacional de Investigación y Desarrollo Tecnológico (Cenidet)

Automated Restructuring System

Log In

Sign Up

Create account

CURP:

NAME:

LAST NAME:

PASSWORD:

CONFIRM
PASSWORD:

USER NAME:

Ilustración 54. Página Registrar nuevos usuarios.

Al ingresar los datos para generar el alta de nuevos usuarios, se realizan ciertas validaciones, que al no cumplirse se muestran ciertos mensajes, como son el registro exitoso de un nuevo usuario o cuando se intenta registrar un usuario existentes y un mensaje de bienvenida cuando ya se ha ingresado. El usuario debe corregir los errores, de lo contrario no será posible la grabación de los datos.

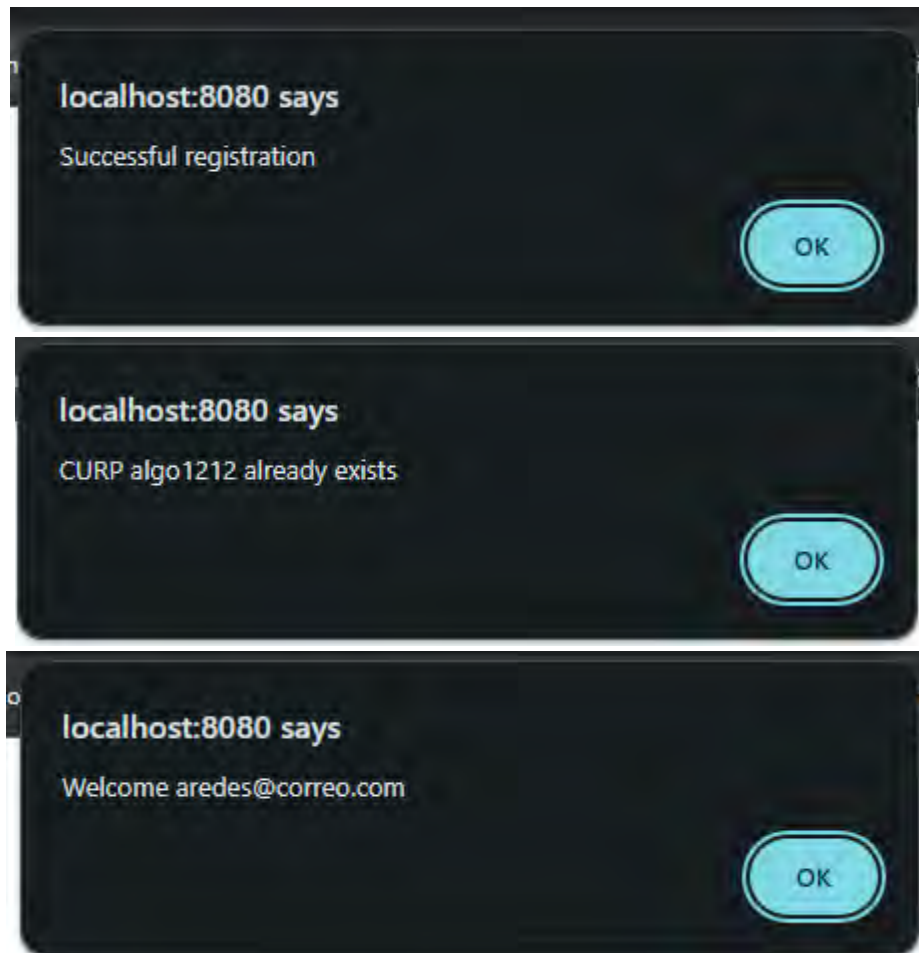


Ilustración 55. Mensajes de notificación de registro de nuevos usuarios.

Error al ingresar al sistema

El siguiente mensaje es presentado cuando el usuario no se encuentra dado de alta en el sistema. Ante este mensaje, deberá seleccionar el comando aceptar y repetir la operación.

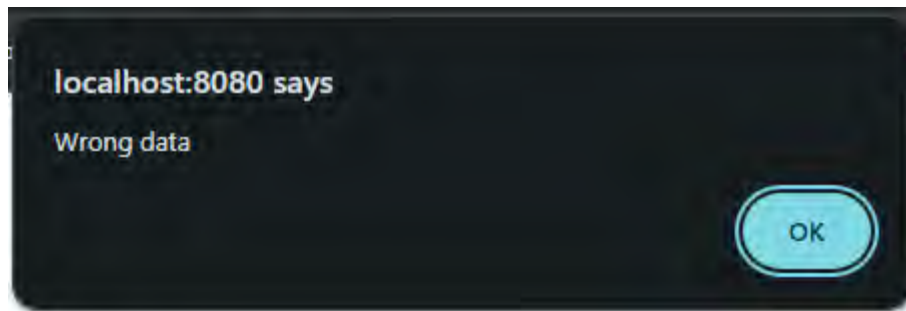


Ilustración 56. Mensaje de ingreso fallido

Menú principal del sistema

Una vez que haya ingresado con éxito al sistema, verá en una barra horizontal las funciones principales de ARE System. En la Ilustración 57 muestra la interfaz gráfica de la aplicación.



Ilustración 57. Menú principal del sistema

Los botones corresponden a las diferentes funcionalidades del sistema, y se explican a continuación

- Botón Upload File (Importar archivo): Permite cargar un archivo en el sistema para ser procesado.

- Botón Evaluate File (Evaluar recurso): Realiza el proceso de evaluación del recurso de aprendizaje contenido en el archivo previamente cargado.
- Botón Check Results (Ver resultados): Muestra los resultados de la evaluación del recurso de aprendizaje.
- Botón Process File (Reestructurar recurso): Lleva a cabo el proceso de reestructuración.
- Botón Download File (Exportar archivo): Permite descargar el archivo con las modificaciones.
- Botón Close Sesion (Cerrar sesión): El sistema finaliza la interacción con el usuario.

5. Cargar recurso de aprendizaje

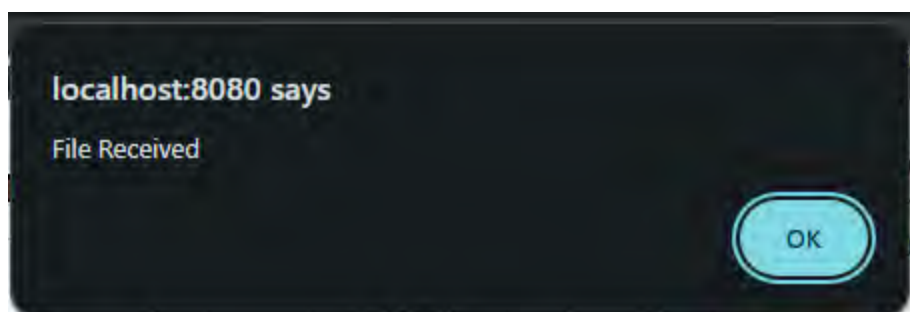
Este comando permite cargar al sistema un archivo PowerPoint que contiene el recurso de aprendizaje, la interfaz de la pantalla se muestra en la Ilustración 58. Seleccione esta opción del menú siempre que desee cargar al sistema un archivo en formato PowerPoint que contiene el recurso de aprendizaje. O bien, si el usuario desea puede seleccionar de la lista de archivos cargados al sistema.



Ilustración 58. Cargar archivos al sistema.

Mensaje de confirmación

Cada vez que el usuario carga un archivo al sistema, éste muestra una notificación indicando que el archivo ha sido recibido por el sistema. En la Ilustración 59 se presenta el mensaje de confirmación. El siguiente mensaje es presentado si la validación es exitosa, ante esto el usuario debe seleccionar el comando aceptar.



6. Evaluar recurso de aprendizaje

El formato de la pantalla para la opción de evaluar recurso se ilustra a continuación. Al ingresar a esta opción se presenta una pantalla en la cual aparece el recurso de aprendizaje original junto al botón para evaluar el recurso de aprendizaje, se muestra en la Ilustración 60.

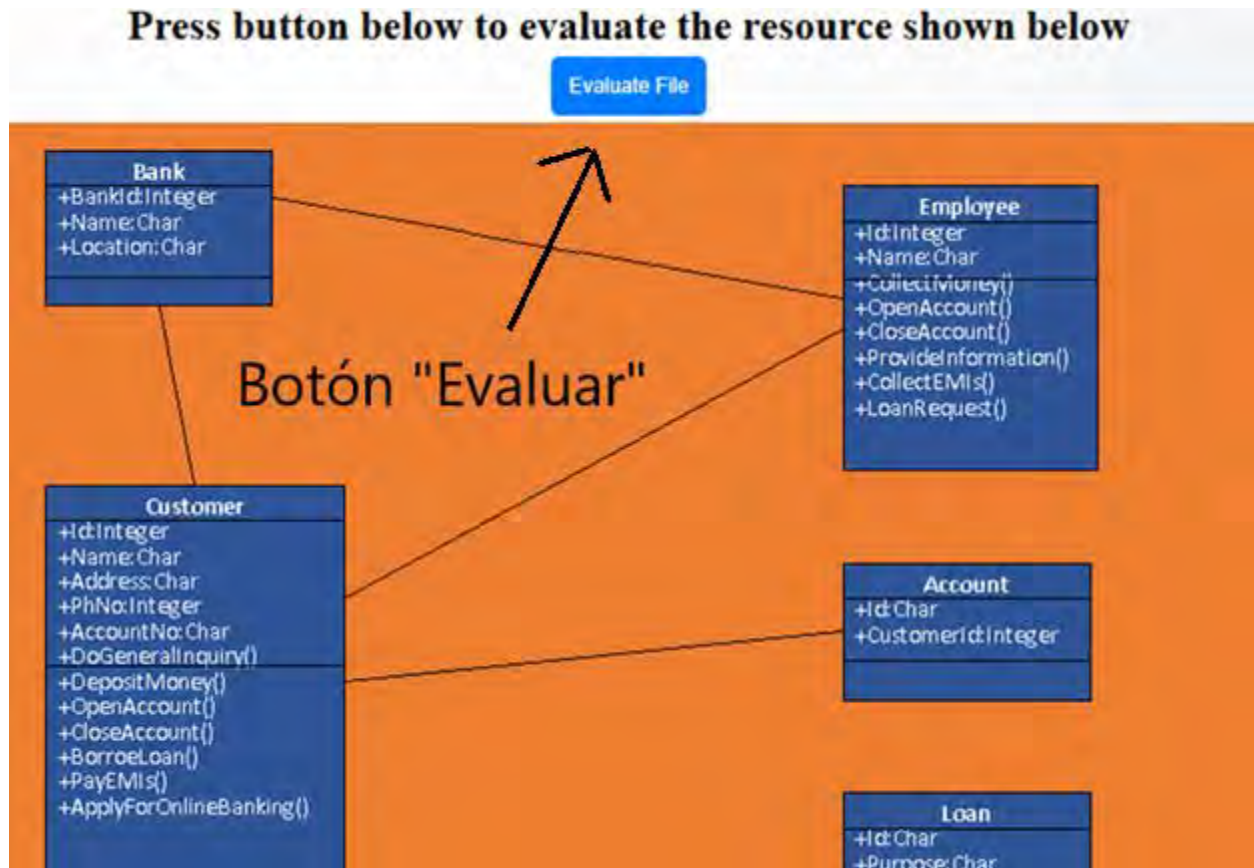


Ilustración 60. Botón "Evaluar Recurso"

Cuando el usuario da clic en el botón Evaluate File, se presenta una notificación indicando que el proceso ha sido completado, el cual se presenta en la Ilustración 61.

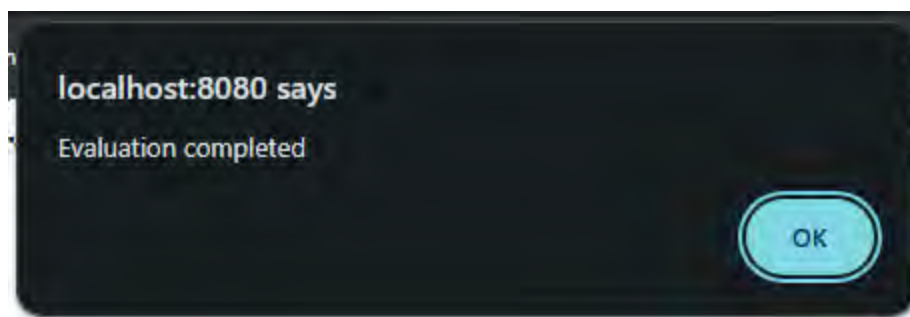


Ilustración 61. Mensaje de evaluación completada

7. Informe de resultados

Seleccione desde el menú principal la opción Check Results o Informe de resultados para ver los resultados del proceso de evaluación. En la Ilustración 62 se muestra una tabla indicando la lista de criterios que determinan la calidad de un recurso de aprendizaje. Los atributos de calidad que si se cumplen se muestran con una casilla de verificación activa, indicando si se cumplió o no con ese atributo, el peso es el valor porcentual de cada atributo y la definición del criterio evaluado, al final la suma de cada atributo contribuye a determinar la correctitud del recurso de aprendizaje.

Evaluation Results		
Weight	Criteria	Check
7%	Slide Background.	☒
14%	Line Color.	☒
10%	Line Width.	☐
8%	Shape width.	☐
7%	Shape color.	☐
10%	Font Size.	☒
10%	Font Color.	☐
34%	Not Blurring.	☒
La evaluación es de 28%		

Ilustración 62. Resultados de evaluación.

8. Procesar recurso de aprendizaje

Seleccionar de la lista proporcionada, los atributos de calidad sobre los cuales el usuario desea reestructurar el recurso de aprendizaje. En la ventana que se muestra en la Ilustración 63, debe indicarse los parámetros de calidad para aplicar los cambios

necesarios. Las opciones marcadas determinan las reglas a aplicar a los recursos de aprendizaje. Una vez seleccionadas las opciones, al presionar el botón 'Process Options' comienza el proceso de reestructura.

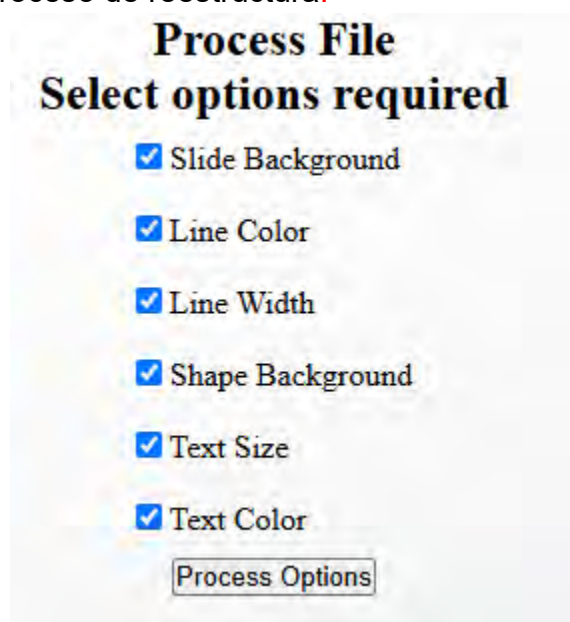


Ilustración 63. Atributos para reestructurar.

Una vez que el proceso ha finalizado, se presenta en pantalla el mensaje de la Ilustración 64, indicando que el recurso de aprendizaje se ha modificado.

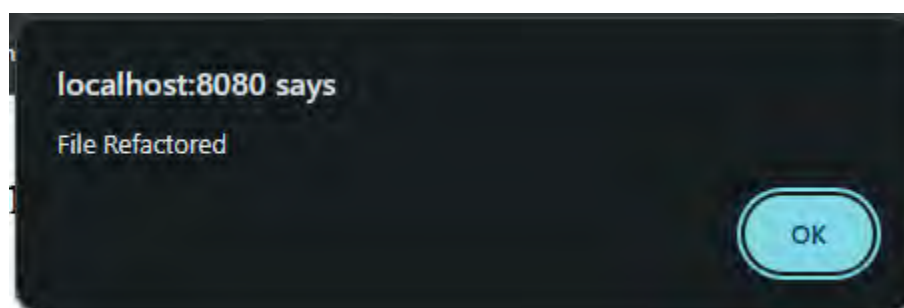


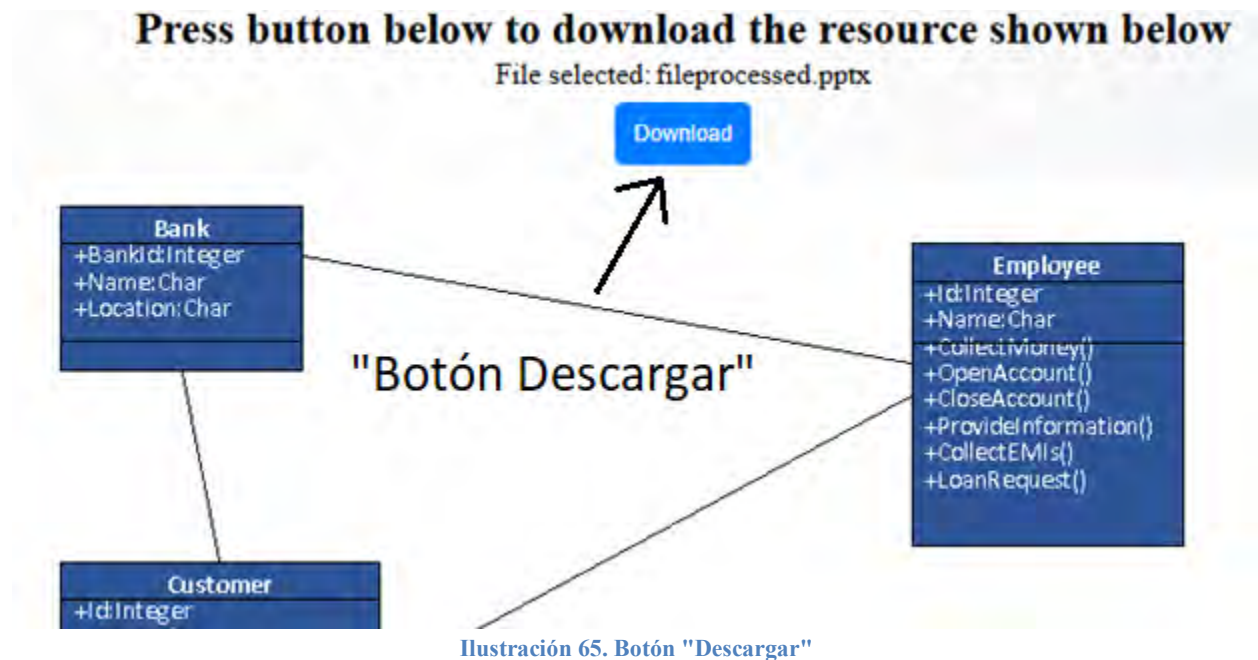
Ilustración 64. Mensaje de confirmación de recurso reestructurado.

9. Exportar recurso de aprendizaje

Esta opción permite crear, a partir del archivo importado previamente, un recurso de aprendizaje reestructurado en formato PowerPoint. El recurso de aprendizaje con los cambios aplicados estará disponible para descargar y al mismo tiempo se podrá visualizar en pantalla, para efectuar la operación se debe seleccionar la opción Download, en ese momento se procede con la descarga del archivo. Los archivos generados por esta opción quedan grabados en el directorio C:\Descargas. Una vez

que ha concluido el proceso de descarga, el usuario podrá iniciar un nuevo proceso para otro recurso de aprendizaje o cerrar sesión.

El archivo PowerPoint que contiene el nuevo recurso de aprendizaje modificado estará disponible en caso de que el usuario desee descargarlo.



10. Salir: al seleccionar esta opción, el sistema retorna a la página principal del sistema.

Anexo 2. Acrónimos

Coq Proof Assistant: welcome. Alloy is an open source language and analyzer for software modeling. It has been used in a wide range of applications, from finding holes in

IEEE: Institute of Electrical and Electronics Engineers

MDE: Model Driven Engineering

MDA: Model Driven Architecture

MDD: Model Driven Development

OMG: Object Management Group

OCL: Object Constraint Language

OOL: A formal Object-Oriented specification Language

PIM: Platform Independent Model

QMOOD: Quality Model for Object-Oriented Design

RDF: Resource Description Framework

SBVR: Semantics of Business Vocabularies and Rules

SDLC: Software Development Life Cycle

TD: Technical Debt

UML: Unified Modeling Language

W3C: World Wide Web Consortium

XML: eXtensible Markup Language

Anexo 3. Resultados de evaluación

Tabla 24. Resultados de evaluación de ARESystem

Rec deAprendizaje	Antes	Después	Rec deAprendizaje	Antes	Después
3Escuela	82	100	52userAdmin	82	100
6vehiculo	72	100	53Membership	82	100
7extintor	61	100	54Estudiante	41	66
8Ariplane	61	100	55Game	100	100
10atm	41	66	56part	100	100
14Banco	82	100	57observer	100	100
15uber	68	100	59pedido	41	66
16Administrativo	76	100	60producto	45	66
18seminar	58	100	61payment	75	100
19books	51	100	62interface	79	100
20banco	65	100	63Persona	100	100
22quiz	42	66	64window	100	100
23Staff	90	100	68spotify	100	100
24order	76	100	72materias	66	66
25window	75	100	74universidad	66	66
27Flight	90	100	79FABULA	66	66
30Blast	72	100	80persona	49	66
31atm	35	66	84package	45	66
32persona	38	66	86transporte	59	66
33Escuela	61	100	88venta	66	66
34buscador	17	66	89almacen	61	100
35empresa	48	66	90redSocial	17	66
36hangout	56	66	95Metrica	45	66
37pedido	100	100	102dieta	59	66
38figura	79	100	103reservacion	59	66
39pedido	24	66	104aereo	93	100
40hospital	27	66	107Inventario	27	66
41hospital	27	66	111Request	93	100
42inventario	56	66	112cuenta	93	100
43biblioteca	42	66	118DBSystem	66	66
44LOGIN	41	100	124Shape	48	66
46conductor	51	100	125Payable	34	66
45menu	24	66	127Payroll	48	66
47hospital	21	66	128Game	31	66
48room	27	66	129GameState	65	100
49task	90	100	130reserva	100	100
50admin	66	66	131Articulo	100	100

Rec deAprendizaje	Antes	Después
132Concurso	68	100
133Biblioteca	48	66
135microservicio	93	100
136Binatang	34	66
137Deportes	68	100
141IA	82	100
146UserNetflix	51	100
147CEO	61	100
148CompanyStructure	75	100
149Telephone	75	100
150TicketSelling	75	100
151Adapter	31	66
152OrdersSystem	82	100
153OnlineShopping	65	100
154Loan	58	100
155OnlineCourse	65	100
156CalendarApp	65	100
157LoanService	82	100
158ShoppingCart	82	100
159Project	82	100
160ProteinModel	48	66
163DesktopItem	41	66
166Assembly	48	66
167CarRental	75	100
168AirlineReservation	75	100
169Inventory	75	100
170Engine	82	100
178ShoppingSystem	93	100
179LibraryMS	72	100
180Components	83	100
181Shape	68	100
183Order	82	100
184Cart	24	66
185TextRepository	58	100
189Course	68	100
192UsingCPU	38	66
201Benchmark	82	100
202BibSonomy	82	100
206Reserve	48	66
207Animal	82	100

Anexo 4. Productos de tesis

Herramienta de Reestructuración para Diagramas de Clase UML Usados como Recurso de Aprendizaje

Areliá Desiderio-Patino*, Olivia Graciela Fragosó-Díaz*, Juan Carlos Rojas-Pérez*

*Departamento de Ciencias de la Computación, TecNM/CENIDET, Cuernavaca, Morelos 62490
México (Tel: 777-362-7770; e-mail: m23ce040@cenidet.tecnm.mx, olivia.fd@cenidet.tecnm.mx,
juan.cp@cenidet.tecnm.mx).

Abstract: The increase in Internet access has allowed students and teachers to make use of learning resources available on the Internet; the daily routine with which this is done has caused them to fall into bad practices such as not verifying the quality of the materials that are used as learning resources. In areas where software development is taught, it is vitally important that the design in early phases of systems development follows the rules to create quality software. Using learning resources that do not comply with software design or quality principles in the initial stages can cause design errors and this results in software with defective quality. This document describes a web tool that makes it easier for the user to restructure UML class diagrams used as learning resources that have previously been evaluated with a quality model. This tool is useful for facilitators or teachers who use UML class diagrams taken from the Internet as learning resources for courses related to software design and development, as well as for people in general in a self-learning process to verify the quality of the resources that they employ.

Keywords: diagramas de clases, legibilidad, recurso de aprendizaje, correctitud, calidad.

1. INTRODUCCIÓN

En los últimos tiempos el internet ha tenido una expansión significativa, incluyendo la conectividad, lo que ha permitido democratizar el acceso a la información. Sin embargo, el que se tenga acceso a una gran cantidad de información, no garantiza que esta sea de calidad. Dentro de internet se pueden encontrar plataformas de aprendizaje, sitios web educativos, entre otros, con una gran variedad de elementos utilizados como recursos de aprendizaje.

La diversidad de fuentes que proveen recursos de aprendizaje en línea incrementa la probabilidad de encontrarse con material de baja calidad o que fueron generados con procesos o prácticas no adecuadas. Utilizar material en línea de manera indiscriminada para procesos de aprendizaje puede causar problemas en la adecuada adquisición de conocimiento a pesar de que dicho proceso este guiado por un facilitador que inadvertidamente este empleando recursos de baja calidad.

Dentro de la Ingeniería de Software, desde etapas tempranas del desarrollo de software, los diagramas UML son ampliamente utilizados entre otras cosas, para especificar y documentar las partes de un sistema de software, por lo que garantizar la correctitud de los diagramas UML contribuye a que en los procesos subsecuentes no se dispersen errores que puedan provocar fallos aún mayores en las etapas finales de desarrollo.

Como etapas tempranas tenemos el análisis y diseño del desarrollo de software, donde se utilizan los diagramas de clase UML para modelar la estructura del sistema y la relación entre sus componentes. Es relevante que desde un inicio se consideren evaluaciones objetivas en cuanto a

calidad durante el proceso de desarrollo de software, buscando que al final se produzca un producto de calidad.

La importancia que los futuros desarrolladores tengan una percepción de calidad en su proceso de formación requiere que los recursos de aprendizaje empleados en su proceso de aprendizaje también cumplan con un nivel de calidad, ya que esto se verá reflejado en su etapa como profesional.

El problema que se aborda es sobre el uso indiscriminado de recursos de aprendizaje obtenidos en línea, específicamente diagramas de clase UML que pudiendo ser de baja calidad, pueden evitar la adecuada adquisición de conocimiento y como consecuencia no lograr los objetivos de aprendizaje esperados.

Para atender el problema antes mencionado, se propone una herramienta web que permite reestructurar diagramas de clases UML utilizados como recursos de aprendizaje. Los atributos que la herramienta puede reestructurar están de acuerdo con un modelo de calidad.

2. TRABAJOS RELACIONADOS

Esta sección trata sobre herramientas de reestructuración de diagramas de clases, mientras que en una sección posterior se presenta un panorama de los parámetros para evaluar la calidad de los recursos de aprendizaje.

2.1 Herramientas de reestructuración

Una propuesta de herramienta de reestructuración se muestra en Moghadam et al (2024) utilizando un enfoque doble, ya que la reestructuración se lleva a cabo tanto en el código como en el modelo UML, considerando que ciertas cosas son más

Sistema de Evaluación de Diagramas de Clase UML Usados como Recurso de Aprendizaje

Desiderio-Patiño Arelia ^{*1}, Fragoso-Díaz Olivia Graciela ¹, Rojas-Pérez Juan Carlos ¹

¹ Departamento de Ciencias de la Computación, TecNM/CENIDET, Cuernavaca, Morelos 62490 México.

* Autor de correspondencia. m23ce040@cenidet.tecnm.mx

Resumen. Cada vez más estudiantes y profesores tienen acceso a una amplia gama de recursos educativos en Internet. Sin embargo, la facilidad con la que se obtiene esta información ha generado la tendencia a no verificar la calidad de los materiales utilizados en el contexto educativo. En el área del desarrollo de software, es esencial que desde las fases iniciales del diseño se priorice la calidad para garantizar la creación de software robusto y bien estructurado. Con el fin de contribuir a la calidad del software, se desarrolló una herramienta web que permite evaluar la calidad de determinados recursos de aprendizaje. Esta herramienta utiliza un modelo de calidad como referencia, asegurando que los materiales empleados en cursos relacionados con el diseño y desarrollo de software cumplan con los estándares de calidad en la ingeniería de software. De esta manera, se previenen errores que podrían comprometer el desarrollo de sistemas desde las primeras etapas.

Palabras clave. Diagramas de clases, recurso de aprendizaje, powerpoint, xml.

I. INTRODUCCIÓN

La evaluación de la calidad de los diagramas de clases UML utilizados como recursos de aprendizaje es fundamental para garantizar su efectividad en la enseñanza de conceptos de programación orientada a objetos. Estos diagramas de clases UML son una herramienta clave para representar la estructura y las relaciones de los componentes de un sistema, sirviendo como una guía visual en las fases iniciales de diseño. Sin embargo, cuando estos diagramas se utilizan como recursos de enseñanza sin una revisión adecuada de su calidad, pueden inducir a errores de comprensión y diseño, afectando negativamente el desarrollo de software robusto y bien estructurado.

En este sentido, la evaluación de la calidad de estos diagramas es crucial para garantizar que cumplan con los estándares de diseño y principios de ingeniería de

software. Un proceso riguroso de evaluación ayuda a identificar posibles deficiencias, como la falta de coherencia, la ambigüedad en la representación de relaciones, o el uso incorrecto de elementos del lenguaje UML. De este modo, la verificación de la calidad no solo asegura la correcta transmisión de conocimientos, sino que también contribuye a la formación de estudiantes y profesionales que siguen buenas prácticas en el desarrollo de software.

La percepción de calidad se considera importante para mantener en los docentes y cultivar en los estudiantes un enfoque en la excelencia y la eficiencia, es por ello por lo que para facilitar que docentes y estudiantes puedan tener un contexto de calidad se generó una herramienta web que evalúa diagramas de clases UML utilizados como recursos de aprendizaje.

II. Trabajos relacionados



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

EL TECNOLÓGICO NACIONAL DE MÉXICO A TRAVÉS DEL CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO

OTORGA EL PRESENTE

RECONOCIMIENTO

A

Areliá Desiderio Patiño, Olivia Graciela Fragoso Díaz

Por su invaluable presentación del Artículo:

Herramienta de Reestructuración para Diagramas de Clase UML Usados como Recurso de Aprendizaje

En el II Congreso Internacional de Tecnología y Ciencia Aplicada, desarrollado en el Tecnológico Nacional de México/Centro Nacional de Investigación y Desarrollo Tecnológico, TecNM/CENIDET, del 22 al 24 de mayo de 2024.

Cuernavaca, Morelos, 24 de mayo de 2024

DR. ARTURO ERNESTO MARES GARDEA
DIRECTOR DEL CENTRO NACIONAL DE INVESTIGACIÓN
Y DESARROLLO TECNOLÓGICO

DR. ARTURO ABÚNDEZ PLIEGO
COORDINADOR GENERAL
II CITCA



Señal Digital:

Este documento es una copia digitalizada de un documento original. La información contenida en este documento es la misma que la contenida en el documento original. La información contenida en este documento es la misma que la contenida en el documento original. La información contenida en este documento es la misma que la contenida en el documento original.

Registro Nacional de Identificación de Documentos

CITCA000070124

cenidet
Centro Nacional de Investigación
y Desarrollo Tecnológico

CITCA





**EL TECNOLÓGICO NACIONAL DE MÉXICO
A TRAVÉS DEL CENTRO NACIONAL DE INVESTIGACIÓN
Y DESARROLLO TECNOLÓGICO**

OTORGA LA PRESENTE

CONSTANCIA

A

Arelia Desiderio Patiño

Por haber asistido y acreditado el curso: "Revisión de la búsqueda bibliográfica y escritura de tesis",
del 07 al 11 de agosto de 2023, con una duración de 30 horas.

Cuernavaca, Morelos, 18 de octubre de 2023



DPA0257323

<http://constancias.cenidet.tecnm.mx>

DR. ARTURO ERNESTO MARES GARDEA
**DIRECTOR DEL CENTRO NACIONAL DE INVESTIGACIÓN
Y DESARROLLO TECNOLÓGICO**

Sello Digital:

CLS1RgvvPtlLhXG2P71VyWUySakCPvqpcrzbDQ18/8bcW+u1/YgMsaCPWMr+07uv79bL307F+yK19jRZ1Zcv5n
/bR8xagVV7b6mXDejy+DKO++AL89WPC19nUzHdAMkz3g27T26seEn/noSSr8W6eLAFDqkkErodwl.dmfUsmZ+Igy
11Xz2+R1u1UE9vK97axK0iXnP/q4oLlk2bg4+eHvsvWCNkw274xV9Tj30nDyLj4DWUq0F9CQ1cuq0Ex8dkgHJja
nCU1K1/15CvPZpdaSgjdclF18AbMlInG48n4j12QjAsvJhAB918RXBQ98/TLX/P16eLm0Qq723gfo+ZKMS1Eg



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

**EL TECNOLÓGICO NACIONAL DE MÉXICO
A TRAVÉS DEL INSTITUTO TECNOLÓGICO DE ZACATEPEC**

OTORGA EL PRESENTE

RECONOCIMIENTO

A

ARELIA DESIDERIO PATIÑO

POR SU DESTACADA PARTICIPACIÓN CON LA PONENCIA INVITADA:

**“CALIDAD DE DIAGRAMAS DE CLASE COMO RECURSOS
DE APRENDIZAJE”**

EN EL SIMPOSIO DE LA RED MULTIDISCIPLINARIA DE INVESTIGACIÓN Y TECNOLOGÍA
PRODEP, DENTRO DEL MARCO DE LAS JORNADAS DE ACADÉMICAS DE INNOVACIÓN,
TECNOLOGÍA, LIDERAZGO Y SOSTENIBILIDAD 2023, CELEBRADO EL 25 Y 26 DE OCTUBRE

Zacatepec, Mor., 26 de octubre de 2023

LORENZO O. HERNÁNDEZ
DIRECTOR





EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

**EL TECNOLÓGICO NACIONAL DE MÉXICO
A TRAVÉS DEL CENTRO NACIONAL DE INVESTIGACIÓN
Y DESARROLLO TECNOLÓGICO**

OTORGA LA PRESENTE

CONSTANCIA

A

Areliia Desiderio Patiño

Por haber asistido y acreditado el curso: "Análisis estadístico de corpus con R", del 08 al 19 de enero de 2024, con una duración de 30 horas.

Cuernavaca, Morelos, 15 de marzo de 2024



DPAD05924

<http://constancias.cenidet.tecnimex.mx>

DR. ARTURO ERNESTO MARES GARDEA
**DIRECTOR DEL CENTRO NACIONAL DE INVESTIGACIÓN
Y DESARROLLO TECNOLÓGICO**

Sello Digital:

uLzRkC+93ceF)Ehu+pi2ZsFB4QjvazlknrkIdAchvRDiROF/oaZcQc+6TafbaAskL)SWZGy2Qk1F7Zk1Q1gLDMSFD1
hZcF+fy7s9RqH7et33CSWQRL1aFsa31A1140hMyqSpISEY71GNpbJg6H9D8FL13g5Wf8Ex4qH1a157J8T0RC4
EFc3UcCW0yN1p2peTTb/2QWyc1Se9wWKAwd/21dL8rRwLSNcUObey/6A4Au90nNRFS0A3Gp/VUzNsOL/7GXIMP1
DLceKfDB1NNFCy3sXJwraeF1eM2ApFWall7MvVzC1u66eS118llo1ahr3adSzbe116XyaRcyb61R6hwaqYvdBQ)---

cenidet
Centro Nacional de Investigación
y Desarrollo Tecnológico





EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

EL TECNOLÓGICO NACIONAL DE MÉXICO

OTORGA EL PRESENTE

DIPLOMA

A

ARELIA DESIDERIO PATIÑO

POR HABER ACREDITADO

EL DIPLOMADO EN CIENCIA DE DATOS, LLEVADO A CABO DEL 29 DE ABRIL AL 29 DE SEPTIEMBRE DEL 2024, CON UNA DURACIÓN DE 120 HORAS EN MODALIDAD EN LÍNEA.

CIUDAD DE MÉXICO, A 16 DE OCTUBRE DE 2024.



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO

DIRECCIÓN GENERAL


RAMÓN JIMÉNEZ LÓPEZ
DIRECTOR GENERAL



Folio: ylk8mL2gh



EL TECNOLÓGICO NACIONAL DE MÉXICO
A TRAVÉS DEL CENTRO NACIONAL DE INVESTIGACIÓN Y
DESARROLLO TECNOLÓGICO

OTORGA LA PRESENTE

CONSTANCIA

A

Areliá Desiderio Patiño

TecNM/CENIDET

Por haber asistido y acreditado el curso: "Introducción a la minería de Texto e Implementación en LM Studio y NPU a principiantes", impartido a estudiantes y profesores del CENIDET del 13 al 17 de enero de 2025, con una duración de 30 horas.



Cuernavaca, Morelos, 31 de enero de 2025

LORENZO O. HERNÁNDEZ

Director del TecNM/CENIDET

DPA008125
<http://constancias.cenidet.tecnm.mx>

cenidet
Centro Nacional de Investigación
y Desarrollo Tecnológico

Sello Digital:

0xUsh7vF8y1erj2bvtQgLy8Pc0W2N2/7yE6YTNc2475FB5aY3sJ5U1wCe1JkQz8SZK0ccac5oU58aQNa2LmsU/MhmlTxxnRMEUco/9KWhjybVYuL/4b1Q1hV9edBfcU2TayUC90rBf3t2aV7zdQqhfEVm2+SBJ6UdrYX1Jc/TXPt/SzRQxvkZfuuPODjkaac9Ypw6i6KbrZCAfXu9jZ1RA2KJ8Z4hJ9cLaNWfo/B/VnuRXH1GwQWyshtnSLXPFE4R90N/vhDdgXSS3uf7MaMLDoLJAoAXhR0xOKj9lWmb6JH05xTenNzySCZY+1oUuNsZr0UGAJ10mfbvv62t8tJ061Swg==



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO

EL TECNOLÓGICO NACIONAL DE MÉXICO
A TRAVÉS DEL CENTRO NACIONAL DE INVESTIGACIÓN Y
DESARROLLO TECNOLÓGICO

OTORGA LA PRESENTE

CONSTANCIA

A

Areli Desiderio Patiño

CENIDET

Por su valiosa participación en la fase local del Concurso 3+3 Minutos sobre mi tesis (3+3M/T), con la presentación del proyecto "Reestructura de recurso de aprendizaje" en el marco de la Divulgación de la Ciencia y Tecnología a jóvenes.

Llevado a cabo el 31 de marzo de 2025 por el Tecnológico Nacional de México campus Centro Nacional de Investigación y Desarrollo Tecnológico (TecNM/CENIDET).



Cuernavaca, Morelos, 31 de marzo del 2025

LORENZO O. HERNÁNDEZ

Director del TecNM/CENIDET

DPAS0025

<https://www.cenidet.mx/verificar-constancia>

cenidet
Centro Nacional de Investigación
y Desarrollo Tecnológico

Dato Digital:

Este documento es una constancia otorgada por el Tecnológico Nacional de México a Areli Desiderio Patiño, quien participó en la fase local del Concurso 3+3 Minutos sobre mi tesis (3+3M/T) con la presentación del proyecto "Reestructura de recurso de aprendizaje" en el marco de la Divulgación de la Ciencia y Tecnología a jóvenes, llevado a cabo el 31 de marzo de 2025 por el Tecnológico Nacional de México campus Centro Nacional de Investigación y Desarrollo Tecnológico (TecNM/CENIDET).

UNIVERSIDAD DE
GUANAJUATO



La Universidad de Guanajuato
otorga el presente

RECONOCIMIENTO

A:

**Desiderio-Patiño Arella, Fragoso-Díaz Olivia Graciela,
Rojas-Pérez Juan Carlos**

Por haber presentado el Artículo titulado "*Refactorización de Diagramas de Clase UML Utilizados en Estrategias de Enseñanza-Aprendizaje*" en el III Simposio de Sistemas Mecatrónicos SISIMek 2024, celebrado el 14 y 15 de octubre de 2024 en las instalaciones de la División de Ingenierías Campus Irapuato-Salamanca.

Salamanca, Gto. 15 de octubre de 2024


Dr. Felipe de Jesús Torres del Carmen
Presidente del Comité Organizador del SISIMek 2024
Lider del UGTO-CA-220 Diseño e Integración de
Sistemas Mecatrónicos




Dr. Antonio de Jesús Salvatín García
Coordinador del Programa Educativo de Licenciatura en
Ingeniería Mecánica
Integrante del UGTO-CA-21 Diseño y Manufactura
Responsable del Laboratorio de Prototipado Digital y Control

**Certificado Tecnológico Nacional de México INTNB101-001 | Cursos MOOC TecNM
2024**

Areliá Desiderio Patiño

completó exitosamente, obteniendo calificación aprobatoria y el
certificado de terminación tipo Honor Code otorgado en Cursos MOOC
TecNM 2024 para

**Tecnológico Nacional de México INTNB101-001:
Inglés para todos Nivel básico 1**

curso ofrecido por Tecnológico Nacional de México.

Coordinación Nacional de Lenguas Extranjeras y Matemáticas
Tecnológico Nacional de México



Número de ID del certificado: 1eb5926b565541909326ca6f0401e8fd

24 de septiembre de 2024



Red Temática Mexicana
de Ingeniería de Software

CONI
SOFT
2023

Acknowledges:

Areli Desiderio Patiño

For attending

The 11th International Conference in Software Engineering Research and Innovation
(CONISOFT 2023) held at the División de Ciencias e Ingeniería, Universidad de
Guanajuato, November 6-10, 2023 in León, Guanajuato, México.

Dra. Hanna Oktaba
REDMIS Honorary President

Dr. Reyes Juárez Ramírez
CONISOFT 2023 General Chair

