



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO
INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA
LICENCIATURA EN CIENCIAS COMPUTACIONALES

TESIS

SoftLearn: SPOC de Análisis de Sistemas de Software

Para obtener el título de
Licenciada en Ciencias Computacionales

PRESENTA
PDLCC. Alexia Zarate Hernández

Directora
MID. Norma Laura Salazar Viveros

Comité tutorial
Dra. Anilú Franco Árcega
Dra. Verónica Martínez Lazcano
Dr. Manuel Alejandro Ojeda Mises

Pachuca de Soto, Hgo., México., julio, 2025



Universidad Autónoma del Estado de Hidalgo
Instituto de Ciencias Básicas e Ingeniería
School of Engineering and Basic Sciences

Mineral de la Reforma, Hgo., a 27 de junio de 2025

Número de control: ICBI-D/1174/2025

Asunto: Autorización de impresión.

MTRA. OJUKY DEL ROCÍO ISLAS MALDONADO
DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH

Con Título Quinto, Capítulo II, Capítulo V, Artículo 51 Fracción IX del Estatuto General de nuestra Institución, por este medio, le comunico que el Jurado asignado a la egresada de la Licenciatura en Ciencias Computacionales **Alexia Zárate Hernández**, quien presenta el trabajo de titulación **"SOFTLEARN: Spoc de Análisis de Sistemas de Software"**, ha decidido, después de revisar fundamento en lo dispuesto en el Título Tercero, Capítulo I, Artículo 18 Fracción IV; dicho trabajo en la reunión de sinodales, **autorizar la impresión del mismo**, una vez realizadas las correcciones acordadas.

A continuación, firman de conformidad los integrantes del Jurado:

Presidente: Dra. Anilú Franco Árcega

Secretario: Dra. Verónica Martínez Lazcano

Vocal: M.I.D. Norma Laura Salazar Viveros

Suplente: Dr. Manuel Alejandro Ojeda Misses

Sin otro particular por el momento, reciba un cordial saludo.

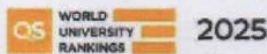
Atentamente
"Amor, Orden y Progreso"

Mtro. Gabriel Vergara Rodríguez
Director del ICBI

GVR/YCC

Ciudad del Conocimiento, Carretera Pachuca-Tulancingo Km. 4.5 Colonia Carboneras, Mineral de la Reforma, Hidalgo, México. C.P. 42184
Teléfono: 771 71 720 00 Ext. 40001
direccion_icbi@uaeh.edu.mx, vergara@uaeh.edu.mx

"Amor, Orden y Progreso"



uaeh.edu.mx

Dedicatoria

A mis padres, por su amor incondicional y por ser mi pilar fundamental. Gracias por su paciencia, comprensión y por enseñarme el valor del esfuerzo. Sus sacrificios y apoyo han sido la base sobre la que he construido este logro. Sin su guía y confianza en mí, no habría podido llegar hasta aquí.

A mi hermano, Juan Pablo, por su apoyo silencioso pero constante, y por recordarme que siempre hay alguien creyendo en mí.

Los amo profundamente y estoy inmensamente agradecido de poder compartir este logro con ustedes. Su compañía y su amor han sido esenciales en este camino, y este triunfo es tan suyo como mío.

A Diego López, por su amor, paciencia y comprensión a lo largo de este largo camino. Gracias por ser mi refugio en los momentos de estrés y por animarme a seguir adelante cuando las cosas se pusieron difíciles.

A mis amigos, por cada palabra de aliento, cada conversación sincera y cada momento compartido.

A mi asesora, Norma Laura Salazar Viveros, por su guía, conocimiento y dedicación. Gracias por su paciencia y por ayudarme a superar los obstáculos que encontré en el camino. Su experiencia y consejos han sido fundamentales para el desarrollo y la culminación de este proyecto.

Finalmente, a todas las personas que, de forma directa o indirecta, han formado parte de este proceso, gracias. Este logro no habría sido posible sin el apoyo y la colaboración de cada uno de ustedes. Gracias por ser parte de este camino y por apoyarme a alcanzar este importante objetivo en mi vida.

Resumen

El presente proyecto “SoftLearn: SPOC de Análisis de Sistemas de Software”, tiene como objetivo diseñar e implementar un curso en línea de tipo SPOC (curso en línea pequeño y privado), enfocado en fortalecer los conocimientos en Análisis de Sistemas de Software.

Este curso está dirigido a estudiantes y egresados que buscan reforzar sus conocimientos, ya sea para cursar la materia de Ingeniería de Software y presentar exámenes extraordinarios.

Para su desarrollo se utilizó la metodología ADDIE, estructurando el curso en fases de análisis, diseño, desarrollo, implementación y evaluación. Se emplearon herramientas como Chamilo y HotPotatoes para crear una plataforma interactiva con módulos interactivos, ejercicios prácticos y evaluaciones automatizadas.

Los resultados muestran una mejora en la comprensión de los temas, así como una alta aceptación por parte de los usuarios. En conclusión, SoftLearn es una herramienta educativa efectiva y accesible, que contribuye al aprendizaje autónomo, refuerza conocimientos clave.

Abstract

The present project, SoftLearn: SPOC for Software Systems Analysis, aims to design and implement an online course in the form of a SPOC (Small Private Online Course), focused on strengthening knowledge in Software Systems Analysis.

This course is intended for students and graduates who seek to reinforce their understanding, whether to successfully complete the Software Engineering course and pass extraordinary exams.

The course was developed using the ADDIE methodology, organizing the process into phases of analysis, design, development, implementation, and evaluation. Tools such as Chamilo and HotPotatoes were used to build an interactive platform with thematic modules, practical exercises, and automated assessments.

The results show improved understanding of key topics and high user acceptance. In conclusion, SoftLearn is an effective and accessible educational tool that supports autonomous learning and reinforces essential knowledge.

Índice general

Introducción	12
Planteamiento del problema	13
Justificación	13
Propuesta de solución	14
Antecedentes.....	15
Objetivo general	16
Objetivos específicos.....	16
Alcance y limitaciones	17
CAPÍTULO I. Marco teórico	18
1.1 Ciencias Computacionales y el desarrollo de software educativo	18
1.1.1 Ciencias Computacionales	18
1.1.2 Aplicación de la Ingeniería de Software en el desarrollo educativo digital	19
1.2 Modalidades del aprendizaje en línea	19
1.2.1 Tipos de cursos en línea: MOOC, SPOC, NOOC.....	19
1.3 Sistemas de Gestión de Aprendizaje (LMS)	22
1.3.1 Evolución de los LMS	22
1.3.2 Características y beneficios de los LMS en la educación digital.....	23
1.3.3 Comparativa de plataformas LMS	25
1.3.4 Comparativa entre LMS y CMS	26
1.3.5 Estándares para garantizar la calidad y compatibilidad en entornos virtuales de aprendizaje.....	26
1.4 Herramientas digitales utilizadas en el diseño del curso	28
1.4.1 Chamilo.....	28
1.4.2 Canva	30
1.4.3 HotPotatoes V.7.....	30
CAPÍTULO II. Marco metodológico	32
2.1 Modelo ADDIE	32
2.1.1 Características de ADDIE.....	32
2.1.2 Fases de ADDIE	33
2.1.3 Ventajas de ADDIE.....	34
2.2 Modelo ASSURE	35
2.2.1 Características de ASSURE	35
2.2.2 Fases de ASSURE	35

2.2.3 Ventajas de ASSURE	37
2.3 Modelo SAM (<i>Successive Approximation Model</i>)	37
2.3.1 Características de SAM	37
2.3.2 Fases de SAM	37
2.3.4 Ventajas de SAM	38
2.4 Modelo Gagné	39
2.4.1 Características de Gagné	39
2.4.2 Fases de Gagné	39
2.4.3 Ventajas de Gagné	41
2.5 Comparación de modelos	41
2.6 Elección de la metodología a implementar	43
CAPÍTULO III. Estado del arte.....	44
3.1 Introducción a la Ingeniería del Software (UAMx - edX)	44
3.2 Software Engineering (Quizlet)	45
3.3 Ingeniería de Software y Arquitectura de Software (Udemy)	46
3.4 Curso de Ingeniería de Software (NobleProg)	47
3.5 Análisis comparativo de herramientas de aprendizaje.....	48
CAPÍTULO IV. Desarrollo del proyecto SoftLearn	50
4.1 Análisis	50
4.1.1 Definición de objetivos.....	50
4.1.2 Análisis del público objetivo	51
4.1.3 Análisis pedagógico.....	51
4.1.4 Análisis técnico.....	52
4.1.5 Cronograma de trabajo	53
4.2 Diseño	53
4.2.1 Diseño pedagógico	54
4.2.2 Diseño de la interfaz de usuario	54
4.2.3 Funcionamiento y elementos de SoftLearn.....	55
4.3 Desarrollo	60
4.3.1 Contenido educativo	60
4.3.2 Ejercicios interactivos en SoftLearn.....	65
4.3.3 Recursos adicionales.....	71
4.4 Implementación	73
4.4.1 Despliegue de la plataforma y configuración del curso.....	73
4.5 Evaluación	77
3.5.1 Análisis de resultados SUS.....	79
3.5.2 Cálculo del puntaje SUS	84

Conclusiones 87

Trabajos futuros..... 88

Referencias 89

Índice de figuras

Figura 1.1. Interacción de roles en un LMS	22
Figura 1.2. Interfaz del apartado de “Administración”	29
Figura 1.3. Interfaz Canva	30
Figura 1.4. Interfaz principal de HotPotatoes.....	31
Figura 2.5. Fases del modelo ADDIE	33
Figura 2.6. Fases del modelo ASSURE.....	36
Figura 2.7. Fases de modelo SAM	38
Figura 2.8. Modelo Gagné	40
Figura 3.9. Interfaz del recurso “Introducción a la Ingeniería del Software”	44
Figura 3.10. Interfaz del recurso “Software Engineering Study Guide”	45
Figura 3.11. Interfaz del recurso “Ingeniería de Software y Arquitectura de Software”	46
Figura 3.12. Interfaz del recurso “Curso de Ingeniería de Software”	47
Figura 4.13. Diagrama Gantt sobre el desarrollo del curso	53
Figura 4.14. Bosquejo de interfaz del curso en Chamilo	54
Figura 4.15. Taxonomía del curso SoftLearn.....	56
Figura 4.16. Portada de la interfaz SoftLearn	61
Figura 4.17. Interfaz Descripción del curso	62
Figura 4.18. Taxonomía del curso con lecciones, ejercicios y la evaluación.	63
Figura 4.19. Interfaz del Módulo 1: Análisis de Sistemas de Software.	63
Figura 4.20. Interfaz de la Lección 1.1: Tipos de requerimientos.	64
Figura 4.21. Evaluación del Módulo 1: Análisis de Sistemas de Software.....	64
Figura 4.22. Ejemplo de ejercicio de “Respuesta única”	65
Figura 4.23. Ejemplo de ejercicio de “Respuesta múltiple”	66
Figura 4.24. Ejemplo de ejercicio de “Rellenar blancos”	66
Figura 4.25. Ejemplo de ejercicio de “Relacionar”.	67
Figura 4.26. Ejemplo de ejercicio de “Coincidencia arrastrable”	67
Figura 27. Ejemplo de ejercicio tipo JQuiz.....	68
Figura 28. Ejemplo de ejercicio tipo JMatch.	68
Figura 4.29. Ejemplo de ejercicio tipo JCross.....	69
Figura 4.30. Ejemplo de ejercicio tipo JCloze.....	70
Figura 4.31. Ejemplo de Ejercicio tipo JMix.	70
Figura 4.32. Interfaz del apartado de “Glosario”	71
Figura 4.33. Interfaz principal del apartado de “Foros”	71
Figura 4.34. Interfaz del apartado de “Enlaces.....	72
Figura 4.35. Interfaz del apartado de “Documentos”.	72
Figura 4.36. Captura de pantalla de la carpeta htdocs.	73
Figura 4.37. Captura de pantalla de la configuración sobre la base de datos local. .	74
Figura 4.38. Panel de control de HostGator	75
Figura 4.39. Conexión del servidor desde FileZilla	75
Figura 4.40. Importación de base de datos “soflear_chamilo”	76

Figura 4.41. Interfaz del dominio SoftLearn.....	77
Figura 4.42. Encabezado de la encuesta.	78
Figura 4.43. Muestra de preguntas primera sesión.	78
Figura 4.44. Gráfico de resultados pregunta 1	79
Figura 4.45. Gráfico de resultados pregunta 2	80
Figura 4.46. Gráfico de resultados pregunta 3	80
Figura 4.47. Gráfico de resultados pregunta 4	81
Figura 4.48. Gráfico de resultados pregunta 5	81
Figura 4.49. Gráfico de resultados pregunta 6	82
Figura 4.50. Gráfico de resultados pregunta 7	82
Figura 4.51. Gráfico de resultados pregunta 8	83
Figura 4.52. Gráfico de resultados pregunta 9	83
Figura 4.53. Gráfico de resultados pregunta 10	84
Figura 4.54. Interpretación de resultados SUS.....	85
Figura 4.55. Evaluación de resultados SUS	86

Índice de tablas

Tabla 1.1. Tabla comparativa de plataformas LMS	25
Tabla 1.2. Tabla comparativa de LMS y CMS	26
Tabla 2.3. Tabla comparativa de las metodologías para el desarrollo de cursos	42
Tabla 3.4. Tabla comparativa de cursos de aprendizaje en línea.....	48

Introducción

La educación actualmente se encuentra en una transformación constante, especialmente por la creciente digitalización del mundo. En este contexto, las metodologías de enseñanza tradicionales resultan insuficientes para responder a las necesidades de las nuevas generaciones de estudiantes, quienes requieren entornos de aprendizaje más dinámicos, accesibles y flexibles (Sosa Díaz & Valverde Berrocoso, 2022).

Ante esta situación, los cursos en línea se han consolidado como una solución eficaz para fomentar el aprendizaje ubicuo (Flores Tena y otros, 2021). Estas propuestas digitales facilitan un estudio autónomo, continuo y práctico, lo que permite integrar retroalimentación inmediata y superar así las limitaciones de los modelos educativos convencionales.

Las ciencias computacionales desempeñan un papel fundamental en la evolución de las plataformas de aprendizaje en línea y los Sistemas de Gestión del Aprendizaje (*Learning Management System* - LMS), ya que los profesionales en este campo no solo se enfocan en la innovación de tecnologías, sino que también proporcionan las bases tecnológicas necesarias para su desarrollo, estas tecnologías permiten organizar, estructurar y distribuir contenidos de forma eficiente, facilitando la enseñanza en diversas áreas del conocimiento (Díaz Quilla y otros, 2021).

En el caso específico del Análisis de Sistemas de Software, uno de los desafíos es la falta de recursos didácticos interactivos que refuercen el aprendizaje práctico. La educación tradicional se centra en contenidos teóricos y evaluaciones estáticas, sin ofrecer suficientes oportunidades para aplicar los conocimientos en contextos reales.

Según Muñoz y Gasca Hurtado (2023) es necesario transformar los procesos educativos para fortalecer la formación en estándares de calidad en el desarrollo de software.

En respuesta a esta situación, las herramientas educativas digitales han demostrado ser una solución efectiva. Estas herramientas no solo facilitan la comprensión de conceptos complejos, sino que también permiten la práctica constante y brindar retroalimentación inmediata (Coaguila Mayanaza y otros, 2023).

La retroalimentación continua se reconoce como un elemento clave para fortalecer la retención del conocimiento y la motivación de los estudiantes, favoreciendo un aprendizaje más interactivo (Arizala Campo y otros, 2023).

Planteamiento del problema

Los avances tecnológicos y el fácil acceso a la información han generado cambios en la manera en que las nuevas generaciones adquieren conocimiento. Acostumbrados a entornos digitales que promueven la inmediatez, la interactividad y la flexibilidad, surge la necesidad de adaptar los modelos educativos a enfoques más dinámicos y orientados al aprendizaje autónomo.

Sin embargo, la enseñanza del Análisis de Sistemas de Software en muchas instituciones de educación superior continúa basándose en métodos tradicionales, como clases magistrales, lecturas teóricas y evaluaciones escritas. Estos enfoques limitan la participación del alumnado y disminuyen las posibilidades de aplicar los conocimientos en escenarios reales.

Una problemática central es la ausencia de recursos interactivos que permitan reforzar los contenidos mediante ejercicios prácticos y evaluaciones constantes. Además, la falta de retroalimentación inmediata dificulta que los estudiantes identifiquen errores a tiempo, afectando la comprensión de conceptos clave lo cual puede disminuir la motivación.

También se observa una escasez de herramientas digitales para el área de Análisis de Sistemas de Software que promuevan el aprendizaje autónomo y faciliten una preparación efectiva, tanto durante el curso de la asignatura de Ingeniería de Software como en evaluaciones extraordinarias.

La ausencia de un entorno de aprendizaje flexible, con actividades automatizadas y acceso constante a los contenidos, representa un obstáculo para la formación de los estudiantes en esta área.

Justificación

La materia de Ingeniería de Software, impartida en los primeros semestres de la Licenciatura en Ciencias Computacionales (LCC), forma parte de los conocimientos esenciales para la formación profesional de los estudiantes.

Sin embargo, conforme avanzan en su trayectoria académica, es común que algunos conceptos fundamentales se desvanezcan con el tiempo, lo cual impacta negativamente en su desempeño en evaluaciones clave.

A pesar de la disponibilidad de recursos en línea sobre Ingeniería de Software, muchos de estos se encuentran dispersos por la red, desactualizados o carecen de

estructura didáctica, lo que dificulta el acceso a contenidos adecuados para el estudio independiente.

A esto se suma la limitada disponibilidad de textos especializados en bibliotecas, lo que restringe las oportunidades de aprendizaje autónomo y práctico para los estudiantes.

En este contexto, se desarrolló SoftLearn, una herramienta digital orientada a fortalecer la enseñanza del Análisis de Sistemas de Software. Este recurso consiste en un curso en línea de tipo SPOC (*Small Private Online Course*), organizado en módulos temáticos que integran cuestionarios, actividades prácticas y evaluaciones automatizadas.

Su diseño permite a los estudiantes repasar y aplicar conceptos clave dentro de un entorno interactivo que favorece el aprendizaje autónomo. SoftLearn se presenta como una opción accesible y especializada para el Análisis de Sistemas de Software, en contraste con otras herramientas educativas que suelen ser costosas o demasiado genéricas.

Su enfoque pedagógico, centrado en el aprendizaje significativo, autónomo y práctico, junto con su implementación tecnológica mediante plataformas accesibles e interactivas, permite cubrir el vacío existente en la enseñanza de esta área, proporcionando un recurso eficiente, bien estructurado y alineado con las necesidades del contexto educativo actual.

Propuesta de solución

En respuesta a las limitaciones de la enseñanza tradicional en el área de Ingeniería de Software, se desarrollará SoftLearn, un curso en línea de tipo SPOC, diseñado para reforzar los conocimientos en Análisis de Sistemas de Software. Este curso ofrecerá a los estudiantes un entorno de aprendizaje estructurado, accesible y dinámico.

Contará con una interfaz intuitiva, desarrollada sobre una plataforma LMS, que permitirá el acceso a módulos temáticos, cuestionarios, ejercicios interactivos y materiales de consulta desde diversos dispositivos. Esto garantizará disponibilidad multiplataforma, accesibilidad remota y continuidad académica asincrónica.

Además, se integrará un sistema de evaluación automatizada con retroalimentación inmediata, lo cual permitirá a los usuarios realizar un seguimiento de su progreso. Esta función no solo facilitará la detección de errores, sino también identificar áreas de

mejora, fortalecer el entendimiento de conceptos e incrementar la motivación de los estudiantes.

El contenido del curso incluirá actividades alineadas con los temas de Análisis de Sistemas de Software, y será útil para reforzar los conocimientos durante el curso de la asignatura de Ingeniería de Software, así como apoyar la preparación en exámenes extraordinarios relacionados con esta misma área.

Antecedentes

La educación ha experimentado una transformación significativa con la incorporación de las Tecnologías de la Información y la Comunicación (TIC), lo que permitió la creación de nuevas estrategias de enseñanza orientadas al aprendizaje autónomo, flexible e interactivo (Flores Tena y otros, 2021).

Estas herramientas han ampliado el acceso al conocimiento, apoyando el desarrollo de habilidades digitales y permitiendo que los estudiantes aprendan a su propio ritmo.

En el ámbito de Ingeniería de Software, el uso de herramientas digitales resulta clave para garantizar que los estudiantes no solo adquieran conocimientos teóricos, sino que también desarrollen habilidades y apliquen sus conocimientos a contextos reales (Pressman, 2010).

La enseñanza de Ingeniería de Software en instituciones de educación superior ha seguido un enfoque tradicional, en el que los docentes imparten contenido teórico mediante clases magistrales y el uso de materiales estáticos, como presentaciones electrónicas o libros impresos y digitales (Muñoz Mata & Gasca Hurtado, 2023).

Sin embargo, estudios recientes indican que este enfoque limita la participación del estudiante y reduce la aplicación práctica de conceptos, afectando la preparación para el entorno laboral real (Torres Vivar y otros, 2024).

La Ingeniería de Software es una disciplina fundamental dentro del campo de las Ciencias Computacionales, ya que proporciona los principios teóricos y metodológicos para el análisis, diseño, desarrollo y mantenimiento de aplicaciones informáticas (Sommerville, 2011).

Además, no solo se enfoca en la creación de software, sino también en la gestión de proyectos y equipos de desarrollo, asegurando que las soluciones tecnológicas sean eficaces, escalables y mantenibles.

El uso de entornos virtuales de aprendizaje, en especial los Sistemas de Gestión del Aprendizaje (LMS), ha demostrado un impacto positivo en la motivación y el desempeño académico de los estudiantes en áreas técnicas, al permitir la organización estructurada del contenido, el aprendizaje autónomo y la evaluación automatizada (Mayorga-Ases y otros, 2024).

Finalmente, la experiencia durante la pandemia por Covid-19 evidenció la relevancia de integrar tecnologías educativas en todos los niveles de enseñanza. La transmisión obligada hacia la educación virtual reveló las deficiencias del modelo tradicional y resaltó la urgencia de fortalecer el uso de herramientas digitales de manera continua, esto es especialmente en áreas como Ingeniería de Software, donde el desarrollo de habilidades prácticas es clave para el desempeño profesional (Pozo y otros, 2024).

Objetivo general

Desarrollar un SPOC (*Small Private Online Course*) orientado al fortalecimiento del aprendizaje en el área de Análisis de Sistemas de Software, utilizando la metodología ADDIE para integrar contenidos modulares, actividades y evaluaciones automatizadas mediante programas de código abierto como Chamilo y HotPotatoes.

Objetivos específicos

Para alcanzar el objetivo general planteado, se han definido objetivos específicos que guiarán el diseño, desarrollo e implementación del curso en línea SoftLearn.

- I. Definir los temas y conceptos clave del Análisis de Sistemas de Software en el área de Ingeniería de Software, basándose en fuentes confiables para estructurar el curso de manera precisa y relevante, con el fin de garantizar un aprendizaje sólido y acorde a las necesidades de los estudiantes.
- II. Diseñar la estructura del SPOC, incluyendo la organización de los módulos, la interfaz y la experiencia de usuario, haciendo uso de principios de diseño centrado en el usuario, con el propósito de crear una plataforma intuitiva, fácil de usar y que facilite la navegación dentro del LMS.
- III. Crear cuestionarios interactivos y actividades que evalúen el conocimiento del usuario en los diferentes subtemas del Análisis de Sistemas de Software,

incorporando retroalimentación inmediata, con el fin de reforzar la comprensión de conceptos clave y permita a los usuarios mejorar constantemente.

- IV. Validar la efectividad del curso mediante la aplicación de encuestas basadas en SUS, con el propósito de identificar áreas de mejora en la experiencia y su impacto.

Alcance y limitaciones

El presente apartado describe los aspectos que delimitan el desarrollo y aplicación del proyecto SoftLearn. Este curso en línea interactivo está diseñado para reforzar los conocimientos en Análisis de Sistemas de Software.

El curso se encuentra alojado en una plataforma LMS y ofrece módulos estructurados, con acceso desde distintos dispositivos que cuenten con conexión a internet. Además, cuenta con un sistema de evaluación automatizada que favorece el aprendizaje autónomo y permite identificar áreas de mejora según el progreso de cada uno.

Sin embargo, este proyecto presenta algunas limitaciones. El acceso al curso depende de la disponibilidad de dispositivos electrónicos y una conexión a internet estable, lo cual puede ser un inconveniente para algunos estudiantes que no cuentan con los recursos tecnológicos adecuados.

Asimismo, debido a la evolución de la Ingeniería de Software, será necesario actualizar periódicamente los contenidos del curso para reflejar el material más reciente en el campo.

CAPÍTULO I. Marco teórico

Para contextualizar el desarrollo de este curso en línea, se presentan a continuación los fundamentos teóricos que explican los conceptos esenciales, siendo la base en la que se sustenta la investigación y el desarrollo del presente proyecto.

1.1 Ciencias Computacionales y el desarrollo de software educativo

Las Ciencias Computacionales tienen un papel fundamental en el desarrollo de herramientas digitales para la educación. En especial, la Ingeniería de Software ofrece metodologías y técnicas para la creación de soluciones tecnológicas y escalables que pueden aplicarse en entornos de enseñanza virtual.

A continuación, se exponen los conceptos clave de las Ciencias Computacionales y su relación con el desarrollo de software educativo, enfocándose en su impacto en el desarrollo de cursos.

1.1.1 Ciencias Computacionales

Las Ciencias Computacionales, también conocidas como Informática, son un campo interdisciplinario que combina conocimientos de ciencias de la computación, matemáticas, estadística y otras disciplinas para estudiar y trabajar con datos e información (López Ortega y otros, 2022).

En la actualidad, estas ciencias juegan un papel crucial en la innovación tecnológica impulsando avances en áreas como inteligencia artificial, ciberseguridad, ciencia de datos, telecomunicaciones y el desarrollo de software.

En el ámbito educativo, los profesionales en Ciencias Computacionales tienen la capacidad de diseñar, desarrollar y mantener sistemas de software que mejoran la experiencia de aprendizaje.

Las tecnologías como las plataformas interactivas, los Sistemas de Gestión del Aprendizaje (LMS) y las evaluaciones automatizadas juegan un papel fundamental en el fortalecimiento de competencias en los estudiantes.

1.1.2 Aplicación de la Ingeniería de Software en el desarrollo educativo digital

La Ingeniería de Software se define como el establecimiento y uso de principios de la ingeniería con el objetivo de desarrollar de forma económica software confiable y eficiente que opere en máquinas reales (Pressman, Ingeniería del software UN ENFOQUE PRÁCTICO, 2010).

Esta disciplina, integrada dentro de las Ciencias Computacionales, se enfoca en el análisis, diseño, desarrollo y gestión de sistemas informáticos de alta calidad, respondiendo a necesidades específicas del usuario.

En el contexto educativo, la Ingeniería de Software ha tenido un impacto significativo al desarrollar entornos virtuales de aprendizaje estructurados y funcionales. Su enfoque metodológico y técnico, es posible crear plataformas adaptativas que incorporan contenidos modulares, actividades interactivas y sistemas de evaluación eficientes (Ibarra-Corona & Escudero-Nahón, 2021).

Además, el uso de esta disciplina facilita la implementación de evaluaciones automatizadas, brindando retroalimentación inmediata para ayudar a los estudiantes a corregir errores y consolidar sus conocimientos.

1.2 Modalidades del aprendizaje en línea

El aprendizaje en línea ha crecido enormemente en los últimos años, debido a los avances tecnológicos, la globalización y la demanda de modelos educativos más adaptables, accesibles y orientados al estudiante (Salinas Ibáñez & Agudelo Velásquez, 2022).

Las plataformas digitales han distribuido los materiales educativos, permitiendo el acceso ubicuo, promoviendo el aprendizaje autónomo y asíncrono. Con base a estas transformaciones, han surgido distintas modalidades de cursos en línea, cada uno con sus propias características en cuanto al alcance, duración, nivel de personalización y tipo de público al que se dirige.

1.2.1 Tipos de cursos en línea: MOOC, SPOC, NOOC

Existen diversas modalidades de cursos en línea que se adaptan a diferentes contextos educativos, públicos y objetivos de aprendizaje. Estas modalidades se diferencian principalmente por su alcance, accesibilidad y objetivo (Herrera, 2024).

A continuación, se explican tres de los modelos más conocidos en el ámbito de la educación digital: MOOC, NOOC y SPOC.

- MOOC (*Massive Open Online Course*)

Los Cursos Masivos Abiertos en Línea (MOOC) son programas formativos disponibles de manera gratuita o a bajo costo para cualquier persona con acceso a internet. Su principal característica es la masividad, ya que pueden ser tomados por muchos usuarios simultáneamente sin restricciones de inscripción (Yuan & Powell, 2013).

Suelen estar respaldados por instituciones de prestigio como Harvard, Stanford, Princeton y suelen ofrecer certificación opcional.

El término MOOC fue acuñado por Dave Cormier y Bryan Alexander en 2008 durante el curso “*Connectivism and Connective Knowledge*”, desarrollado por Siemens y Downes en la Universidad de Manitoba.

De acuerdo con Treviño y Irais (2024), algunos ejemplos reconocidos son:

- edX (2023), desarrollada por Harvard y el MIT, ofrece cursos universitarios gratuitos en áreas como informática, ciencia de datos y humanidades.
- Coursera (2012), en alianza con universidades y empresas como Stanford, Yale y Google, proporciona programas especializados y certificaciones profesionales.

Los MOOC combinan materiales audiovisuales cortos (8 a 12 minutos), autoevaluaciones, exámenes, foros de discusión y actividades colaborativas. Algunos emplean evaluación automática y permite aprender a ritmo propio, gracias a su diseño flexible (Ortiz González, 2016).

Esta modalidad ha recibido críticas debido a las altas tasas de abandono, la limitada interacción entre alumnos y docentes y su creciente orientación comercial.

- NOOC (*Nano Open Online Course*):

Los Nano Cursos Abiertos en Línea (NOOC) son una modalidad más reciente enfocados al microaprendizaje, centrados en el desarrollo de una sola habilidad, destreza o área de conocimiento. Suelen tener una duración de entre 1 y 10 horas, aunque puede alcanzar hasta 20 horas, lo que permite que los estudiantes adquieran conocimientos en concreto rápidamente (López de la Serna y otros, 2018).

Este formato resulta útil para el aprendizaje autodirigido, flexible y continuo. Los NOOC fueron creados durante la última década como alternativa al aumento de la demanda por cursos cortos, prácticas y de rápida implementación.

A pesar de su brevedad, los NOOC mantienen los principios de apertura, accesibilidad y flexibilidad que caracterizan los MOOC.

Esta modalidad se fundamenta en una *Nano Learning Experience* (NLE), es decir una experiencia breve pero estructurada que debe incluir: contenido específico, actividades para evidenciar lo aprendido, evaluación de logro obtenidos y acreditación por una insignia digital (Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF), 2025).

Ejemplos conocidos son:

- *LinkedIn Learning* (LinkedIn), que ofrece cursos de corta duración orientados al desarrollo de habilidades profesionales, como liderazgo, programación y productividad.
- *Google Digital Garage* (Google), plataforma educativa gratuita que brinda capacitación en habilidades digitales básicas, marketing en línea y herramientas para negocios.

Aunque estos cursos no se centran en el aprendizaje profundo de una disciplina, permiten a los estudiantes adquirir competencias clave siendo de gran utilidad los usuarios para mejorar sus perfiles.

▪ **SPOC (*Small Private Online Course*):**

Los Cursos Online Pequeños y Privados (SPOC) son una modalidad educativa que surge como una evolución de los MOOC, pensados para grupos reducidos y con acceso restringido. Estos cursos están orientados a un público específico, como empleados de una empresa o estudiantes de una universidad lo que permite ofrecer una experiencia de aprendizaje más personalizada y controlada (Pinilla Acevedo, 2016).

Ejemplos comunes incluyen:

- Cursos institucionales dirigidos únicamente a estudiantes matriculados.
- Programas de formación continua o capacitación corporativa con tutorización y seguimiento.

A diferencia de los MOOC, los SPOC se caracteriza por su enfoque más centrado al estudiante lo que permite un mayor seguimiento de su progreso, debido a esto los hace adecuados para entornos educativos formales, donde la interacción y el seguimiento por parte del instructor son clave. Además, el número de participantes y el acceso al curso están limitados (Aguayo Sarasa & Bravo-Agapito, 2017).

Los SPOC se han vuelto cada vez más relevantes en el ámbito educativo, ya que permite adecuar los contenidos formativos a demandas particulares, tanto

del alumnado como las propias instituciones, permitiendo así una oferta educativa personalizada.

En el ámbito de la educación a distancia, los SPOC han sido utilizados como recursos de refuerzo académico.

Debido a estas características el proyecto SoftLearn se enfoca en el desarrollo de un SPOC, orientado específicamente a la enseñanza del Análisis de Sistemas de Software.

1.3 Sistemas de Gestión de Aprendizaje (LMS)

Los *Learning Management Systems* (LMS) son plataformas digitales que ayudan a planificar, impartir, gestionar y evaluar procesos de enseñanza en línea, mediante un estándar internacional bien organizado que permite el acceso y control de los materiales educativos de forma centralizada (Díaz Quilla y otros, 2021).

Además, los LMS funcionan como software alojado en un servidor web, lo que permite que múltiples usuarios interactúen desde diferentes ubicaciones. Como se muestra en la Figura 1.1, la plataforma se organiza en tres roles administradores, profesores y alumnos. Los administradores gestionan tanto los cursos como la administración general de la plataforma, los profesores acceden al LMS para impartir lecciones, asignar ejercicios y realizar evaluaciones a los estudiantes y los alumnos acceden a los cursos, el contenido y a las actividades.

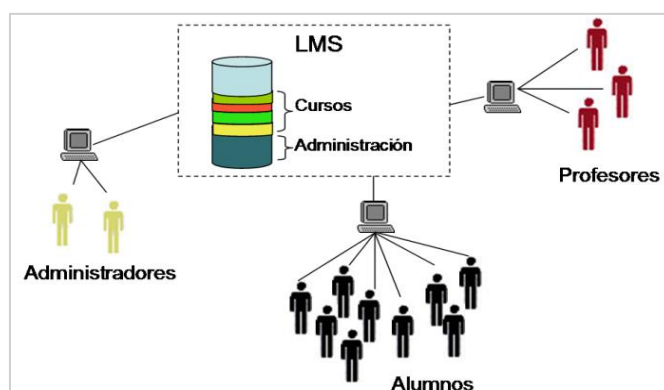


Figura 1.1. Interacción de roles en un LMS (López, 2005).

1.3.1 Evolución de los LMS

Desde sus orígenes en la década de 1990, los LMS se limitaban a páginas HTML estáticas con poca interactividad. Los primeros sistemas virtuales utilizaban herramientas básicas como el correo electrónico, los foros de discusión y actividades en línea desarrolladas con Java o JavaScript (Castro y otros, 2013).

Con la expansión al acceso de internet, la necesidad de plataformas más estructuradas se hizo presente. A medida de los años 90, surgieron las primeras herramientas diseñadas específicamente para la creación y gestión de cursos en línea. Estas plataformas, facilitan la administración de contenido sin la necesidad de conocimientos avanzados en programación o diseño gráfico.

El blog de la Cátedra UNESCO (2013) de Educación a Distancia (CUED) menciona que los LMS han evolucionado de simples repositorios de contenido a sistemas integrales de gestión de aprendizaje.

Principalmente estas plataformas eran utilizadas por universidades y empresas para centralizar la enseñanza, pero a lo largo del tiempo, se convirtieron en soluciones indispensables para la educación.

Un estudio publicado por DELTA INITIATIVE presenta un análisis de la evolución de los LMS desde el año 1997 hasta el 2010, durante este periodo, los LMS pasaron de ser soluciones experimentales a herramientas estandarizadas en el ámbito educativo, marcando el inicio de una nueva era en la enseñanza en línea.

Actualmente, los LMS siguen evolucionando continuamente con el nuevo auge de la inteligencia artificial, analítica de aprendizaje, personalización del contenido y gamificación, elementos que aportan a una experiencia más adaptativa, motivadora y efectiva.

1.3.2 Características y beneficios de los LMS en la educación digital

Los LMS cuentan con características fundamentales que han transformado la forma de impartir el aprendizaje en entornos virtuales. Estas cualidades además de facilitar la administración de contenidos también promueven un aprendizaje más flexible, accesible y centrados a los estudiantes (Castro y otros, 2013) (Lorenzo Garcia, 2020).

Entre las principales características y beneficios se encuentran:

- **Interactividad:**
Capacidad de comunicación bidireccional entre los usuarios y la plataforma. Lo que permite que el estudiante tenga control sobre el nivel de interacción con los recursos.
- **Flexibilidad y accesibilidad:**
Los LMS permite a los alumnos acceder a los cursos desde cualquier lugar, dispositivo y horario, lo que favorece el aprendizaje autónomo.

- **Escalabilidad:**
Capacidad de los LMS para crecer sin comprometer su rendimiento y calidad. Además de poder soportar un gran número de usuarios y cursos sin afectar su funcionalidad
- **Estandarización:**
Los LMS deben seguir normas y estándares para garantizar la interoperabilidad entre plataformas educativas como SCORM, lo que permite importar y exportar contenidos entre diferentes plataformas.
- **Usabilidad:**
Se trata de la facilidad de uso de la plataforma. Deben ser intuitivos, rápidos y de fácil acceso para los usuarios, asegurando un aprendizaje eficiente.
- **Funcionalidad:**
Capacidad de cumplir con los requerimientos y necesidades de los usuarios, ofreciendo herramientas adecuadas para la enseñanza. Su funcionamiento está relacionado con la escalabilidad y la capacidad de adaptación.
- **Personalización y adaptación:**
Permite ajustar los cursos y actividades a las necesidades de los estudiantes, ofreciendo un servicio más personalizado.
- **Gamificación:**
Pueden incorporar elementos de gamificación como puntajes, insignias o tablas de clasificación a los sistemas para hacer el aprendizaje más atractivo y motivar a los estudiantes, lo que lo hace más atractivo.
- **Ubicuidad:**
Capacidad de acceder a los LMS desde cualquier lugar y en cualquier momento. Lo que facilita el aprendizaje autónomo y la disponibilidad de los materiales educativos.
- **Persuasibilidad:**
Capacidad para motivar a los estudiantes a seguir aprendiendo. Es la integración de funcionalidad, usabilidad, interactividad y accesibilidad para mejorar la experiencia.

Gracias a sus características, los LMS se han convertido en herramientas fundamentales para la educación actual, favoreciendo la calidad del aprendizaje, la gestión docente y la experiencia del usuario.

1.3.3 Comparativa de plataformas LMS

En la Tabla 1.1 se comparan cuatro plataformas LMS destacadas, considerando sus licencias, características, principales ventajas y desventajas:

Tabla 1.1. Tabla comparativa de plataformas LMS.

	Chamilo	Moodle	Canvas LMS	Blackboard
Código abierto	Si	Si	No	No
SCORM	Si	Si	Si	Si
Facilidad de uso	Alta	Media	Alta	Media
Personalización	Alta	Alta	Alta	Media
Características	Interacción (foros, chats, redes sociales), gestión de cursos, administración de documentos, interfaz intuitiva	Compatible con SCORM, foros, wikis, encuestas, cuestionarios, adaptable a necesidades específicas	Interfaz moderna, integración con herramientas externas, nube, API robusta.	Alta seguridad, herramientas de evaluación avanzadas, personalización para instituciones
Actividades soportables	Opción múltiple, rellenar espacios, verdadero/falso, asociación, arrastrar y soltar, redacción, foro, chat, encuestas, tareas, wikis, cuestionarios, ejercicios interactivos, lecciones, debates	Opción múltiple, rellenar espacios, verdadero/falso, arrastrar y soltar, encuestas, foros, chat, tareas, lecciones, glosario	Opción múltiple, rellenar espacios, verdadero/falso, encuestas, foros, chat, tareas, lecciones, glosario	Opción múltiple, rellenar espacios, verdadero/falso, simulaciones, encuestas, foros, chat, tareas
Ventajas	Fácil de usar, interfaz intuitiva, multi-idioma, herramientas de videoconferencia	Gran comunidad, modular, adaptable a diferentes necesidades	Interfaz intuitiva, fácil integración con herramientas de terceros, buena experiencia móvil	Alta seguridad, compatible con instituciones educativas grandes, herramientas avanzadas
Desventajas	Requiere tiempo de instalación e implementación	Requiere configuración avanzada, curva de aprendizaje	Requiere conexión a internet constante, puede ser costoso para algunas instituciones	Interfaz menos intuitiva que otras opciones, costos elevados

Fuente: Autoría propia.

1.3.4 Comparativa entre LMS y CMS

En el ámbito de la educación digital, existe dos tipos de plataformas aunque comparten ciertos aspectos, tienen propósitos distintos: LMS y *Content Management Systems* (CMS).

En la Tabla 1.2 se comparan sus principales diferencias en términos de objetivos, funciones y aplicaciones (Cañellas Mayor, 2011):

Tabla 1.2. Tabla comparativa de LMS y CMS.

	LMS	CMS
Propósito principal	Facilitar la gestión del aprendizaje y la enseñanza	Administrar y publicar contenidos digitales
Enfoque	Educación y formación	Administración de contenido web
Interacción con usuarios	Seguimiento del progreso del estudiante, evaluaciones, foros de discusión	Creación, edición y gestión de contenido sin enfoque en el aprendizaje
Herramientas principales	Cursos, evaluaciones, foros, retroalimentación, seguimiento del progreso	Edición de textos, gestión de menús, administración de multimedia, publicaciones jerárquicas
Usuarios	Docentes, alumnos ya administradores educativos	Diseñadores web, editor de contenido, empresas
Ejemplos comunes	Chamilo, Moodle, Canvas, Blackboard	WordPress, Joomla, Drupal
Aplicación en educación	Alta: diseñado para entornos de aprendizaje estructurados	Limitada: puede adaptarse, pero no está optimizado para gestión educativa

Fuente: Autoría propia.

Esta comparación es clave para entender por qué se eligió un LMS en el desarrollo del curso SoftLearn y no un CMS.

Dado que los CMS, se centran en la publicación de contenidos estáticos y los LMS están diseñados específicamente para facilitar la enseñanza y el aprendizaje en entornos digitales.

1.3.5 Estándares para garantizar la calidad y compatibilidad en entornos virtuales de aprendizaje

El correcto funcionamiento de los cursos en línea no solo depende de la plataforma (LMS), sino también del cumplimiento de estándares para garantizar la interoperabilidad, la calidad y la reutilización del contenido en los diferentes entornos de aprendizaje. Entre los más relevantes se encuentra el modelo SCROM y las normas ISO aplicables al software educativo.

- SCORM (Shareable Content Object Reference Model)

Los estándares de interoperabilidad son normas técnicas y especificaciones que aseguran que los recursos educativos sean reutilizados, compartidos y ejecutados correctamente en distintas plataformas LMS. Uno de los más utilizados en el ámbito de e-learning es SCORM.

El Modelo de Referencia para Objetos de Contenido Compartible (SCORM) es un estándar internacional para la creación de contenidos e-learning reutilizables y compatibles con múltiples LMS.

Su principal ventaja es facilitar la distribución y reutilización de los recursos sin necesidad de grandes modificaciones, es un formato estandarizado que puede implementarse fácilmente en cualquier plataforma compatible, gracias a su estructura (Colman, 2024).

SCORM permite la incorporación de cuestionarios, actividades dinámicas y retroalimentación automática dentro de su entorno virtual, garantizando su compatibilidad con Chamilo y otros LMS que siguen este estándar. De igual forma SCORM permite llevar el seguimiento del progreso de los estudiantes, lo que facilita el monitoreo académico (Laguna Lozano, 2024).

- Normas ISO como referencia para el desarrollo de software educativo

Además de los estándares de interoperabilidad, existen marcos normativos que garantizan la calidad del software educativo. Las normas de la Organización Internacional de Normalización (ISO) proporcionan marcos teóricos y metodológicos que orientan la calidad, funcionalidad, eficiencia e interoperabilidad de los productos digitales (Organización Internacional de Normalización (ISO), 2025) .

Aunque no se implementaron de manera certificada, estas normas sirvieron como referencia para establecer buenas prácticas en el diseño y desarrollo del entorno virtual.

A continuación, se presentan las más relevantes:

- ISO/IEC 25010 – modelo de calidad del producto de software: Este modelo evalúa la calidad de los aspectos como la funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad (ISO/IEC, 2023).

Estos criterios son considerados durante la selección de la plataforma y en el diseño del curso, buscando que el entorno sea fácil de usar, accesible y adaptable a las necesidades de los estudiantes.

- ISO/IEC 12207 - procesos del ciclo de vida del software: Define los procesos necesarios del ciclo de vida para el desarrollo de software, desde el análisis hasta el mantenimiento. Este estándar proporciona un marco estructurado para gestionar las actividades y tareas específicas de cada fase del ciclo de vida.

En el desarrollo de software educativo garantiza que los procesos de desarrollo estén claramente definidos y que el software final cumpla con los estándares internacionales de calidad (ISO/IEC/IEEE, 2020).

- ISO/IEC 19796-1 - gestión de la calidad en el aprendizaje, la educación y la formación: Este estándar presenta un método estructurado para la mejora continua los procesos de enseñanza y aprendizaje asegurando la calidad de estos (ISO/IEC, 2018).

Por ejemplo, incorporar retroalimentación automática, evaluación progresiva y criterios de mejora a partir de las pruebas piloto con estudiantes.

Gracias a la integración del estándar SCROM y la aplicación de normas ISO, es posible crear sistemas educativos que garanticen una experiencia de aprendizaje óptima, alineada con los principios de calidad, accesibilidad y satisfacción del usuario.

1.4 Herramientas digitales utilizadas en el diseño del curso

Para el diseño e implementación del curso SoftLearn, se seleccionaron herramientas digitales que facilitan la gestión de contenidos, el diseño visual y la creación de actividades interactivas.

A continuación, se describen las principales herramientas empleadas:

1.4.1 Chamilo

Chamilo es un Sistema de Gestión de Aprendizaje (LMS) de código abierto orientado a la creación y gestión de cursos virtuales. Destaca por su facilidad de uso, flexibilidad y capacidad para escalar según las necesidades institucionales (Chamilo, 2020).

Entre sus principales características se encuentran la organización eficiente de materiales educativos, la integración de contenido multimedia, el desarrollo de actividades interactivas y la administración de foros de discusión.

De igual forma cuenta con un sistema de calificaciones automatizado que permite evaluar a los estudiantes de manera rápida y precisa, ofreciendo de igual manera un seguimiento detallado del progreso y desempeño de cada uno de los estudiantes.

Una de sus principales ventajas es su compatibilidad con el estándar SCORM, lo que la convierte en una plataforma ideal para integrar cursos que cumplan con este estándar, lo que facilita la reutilización e integración de contenidos en diversas plataformas.

En SoftLearn, Chamilo funciona como la plataforma central para estructurar los módulos de enseñanza, gestionar cuestionarios, foros, actividades y proporcionar retroalimentación inmediata.

Como se observa en la Figura 1.2 la interfaz de “Administración”, se muestran algunas de las herramientas y configuraciones para gestionar la plataforma, los usuarios, cursos y otras funciones del sistema.

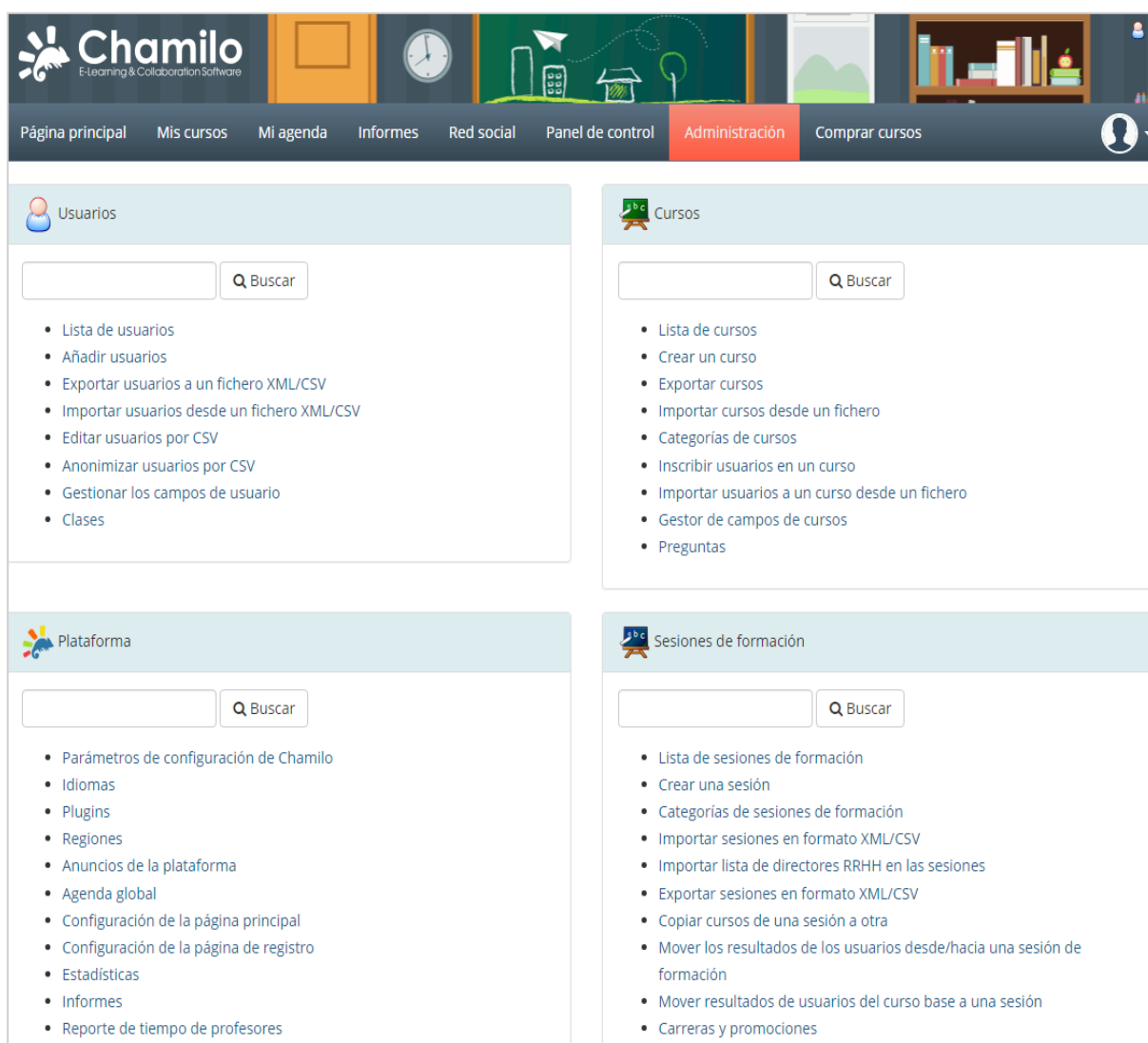


Figura 1.2. Interfaz del apartado de “Administración” (Chamilo, 2020).

1.4.2 Canva

Canva es una herramienta de diseño gráfico en línea que permite a los usuarios crear recursos visuales como infografías, esquemas, mapas conceptuales y presentaciones sin necesidad de grandes conocimientos avanzados (Canva, 2021).

En SoftLearn, se utiliza para desarrollar materiales gráficos que complementan el contenido teórico, facilitando la comprensión de los conceptos de Ingeniería de Software, permitiendo exportar diseños a una gran variedad de formatos, incluidas imágenes (PNG, JPG), documentos pdf e incluso videos, lo que la hace una opción versátil.

Como se muestra en la Figura 1.3 la interfaz de edición, en el cual se aprecia el espacio de trabajo en blanco y el menú de herramientas el cual incluye plantillas para editar o en su caso todos los elementos para crear un diseño.

Cabe señalar que permite la colaboración en tiempo real entre docentes y ofrece almacenamiento en la nube para gestionar los recursos de manera eficiente.

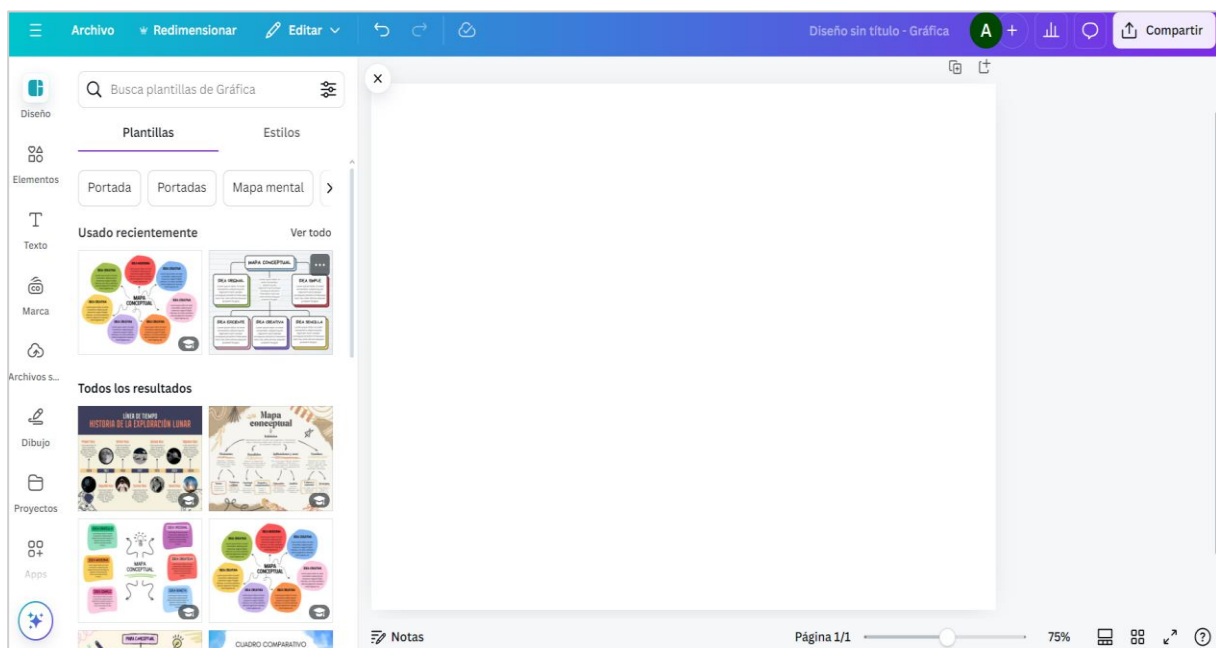


Figura 1.3. Interfaz Canva (Canva, 2021).

1.4.3 HotPotatoes V.7

HotPotatoes es un conjunto de herramientas de software educativo que facilita la creación de actividades interactivas, como crucigramas, sopas de letras, emparejamientos, cuestionarios y preguntas de respuesta corta, todos estos facilitan el aprendizaje de manera atractiva y eficaz (HotPotatoes, 2005).

Como se muestra en la Figura 1.4 es la interfaz principal de HotPotatoes, donde se aprecian y agregaron detalles de todas las diferentes actividades que el usuario puede realizar.

Gracias a su compatibilidad con el estándar SCROM, permite generar materiales didácticos atractivos y compatibles con LMS como Chamilo. En SoftLearn, se utilizan para diseñar actividades de refuerzo dinámicas, proporcionando retroalimentación inmediata y fomentando el aprendizaje activo.

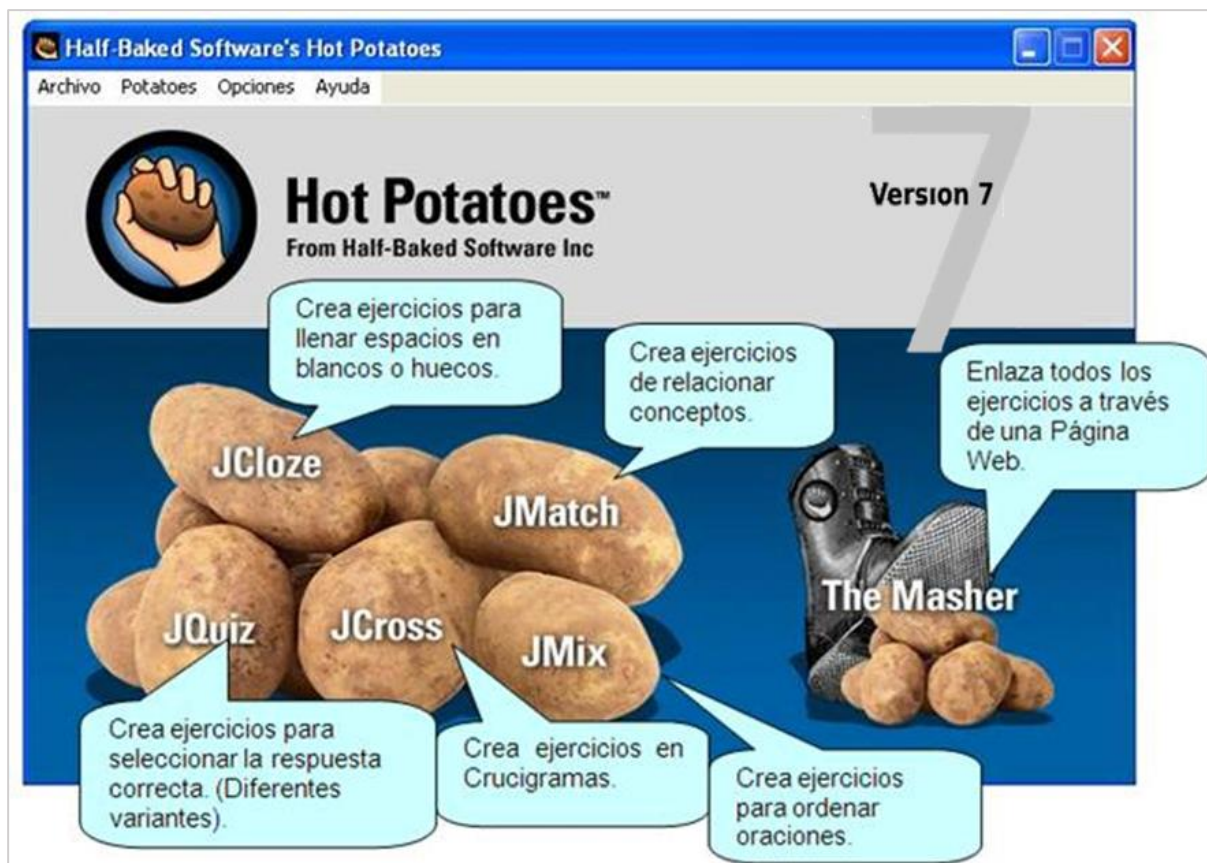


Figura 1.4. Interfaz principal de HotPotatoes (HotPotatoes, 2005).

CAPÍTULO II. Marco metodológico

El diseño de un curso en línea debe basarse en una metodología clara que considere tanto aspectos pedagógicos como tecnológicos.

Desde el enfoque pedagógico, se busca promover un aprendizaje significativo, incentivar la participación de los estudiantes y estar alineado con objetivos formativos específicos. En cuanto al aspecto tecnológico, es fundamental contar con una estructura que garantice la accesibilidad, funcionalidad e interoperabilidad de los recursos educativos, facilitando su implementación en plataformas digitales.

En este capítulo se describen cuatro modelos metodológicos: ADDIE, ASSURE, SAM y Gagné. Posteriormente, se presenta una comparación entre ellos con el fin de escoger la metodología más adecuada para el desarrollo del presente proyecto, SoftLearn.

2.1 Modelo ADDIE

Uno de los modelos más utilizados en el diseño instruccional es el modelo ADDIE (Análisis, Diseño, Desarrollo, Implementación y Evaluación), reconocido por su eficiencia en la planificación y creación de cursos e-learning y programas de formación.

Fue desarrollado en la década de 1970 por el ejército de los Estados Unidos en colaboración con la Universidad de Florida (Defense Technical Information Center, 1975), con el propósito de mejorar la eficiencia y efectividad del diseño instruccional en la formación militar.

2.1.1 Características de ADDIE

Según ISpring (2024), las principales características del modelo ADDIE son:

- **Flexible:** Capacidad de adaptarse a diferentes tipos de programas de capacitación, entornos educativos y enfoques de diseño instruccional.
- **Iterativo:** Sigue un proceso cíclico, permitiendo una retroalimentación y mejora continua, basándose en las opiniones de los participantes.
- **Centrado en las necesidades:** Se enfoca en los objetivos específicos de los estudiantes y la organización, permitiendo que sea relevante y efectivo.

- Colaborativo: Involucra a expertos en la materia, docentes y estudiantes en el proceso de desarrollo, fortaleciendo la calidad del diseño.
- Orientado a metas: Dirigido al cumplimiento de los objetivos propuestos y en la obtención de los resultados deseados

Estas cualidades hacen de ADDIE una metodología completa y efectiva para el diseño instruccional.

2.1.2 Fases de ADDIE

De acuerdo con la Universidad Estatal de Florida, este modelo se desarrolla en cinco fases como se observa en la Figura 2.5:

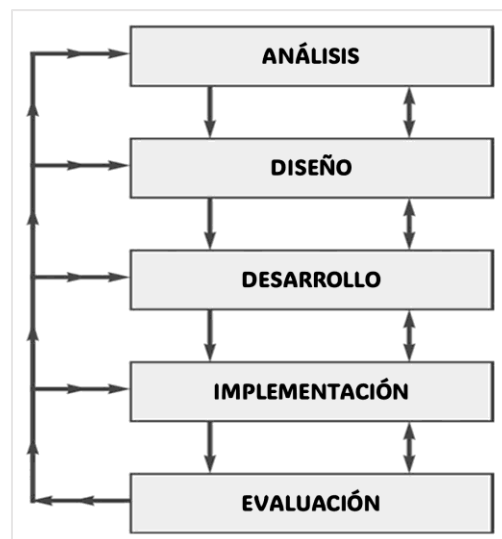


Figura 2.5. Fases del modelo ADDIE (Molenda, 2003).

1. Análisis:

En esta etapa inicial se identifican los problemas de aprendizaje y se buscan las mejores estrategias para solucionarlos. Se hace un estudio de las necesidades educativas, recursos disponibles y las limitaciones del entorno.

Se identifican los objetivos del programa y se determinan la estrategia para impartir los materiales, además de planificar los temas y contenidos en términos generales, y se asignan responsabilidades a los miembros del equipo.

2. Diseño:

En esta etapa se hace uso de los hallazgos del análisis previo para estructurar el contenido y se establecen las estrategias pedagógicas para alcanzar los objetivos de aprendizaje.

Se establecen objetivos de aprendizaje, contenidos a tratar, los métodos de evaluación y los medios tecnológicos a utilizar. El contenido del curso debe garantizar una secuencia lógica que facilite la comprensión del conocimiento.

De igual forma se diseña la interfaz del curso, asegurando que sea intuitiva y accesible para los estudiantes

3. Desarrollo:

Se centra en el desarrollo de los materiales didácticos y la implementación de las herramientas necesarias para el curso.

Se generan los contenidos de los distintos formatos, como textos, imágenes y actividades interactivas, asegurando que sean atractivas y cualquier otro recurso necesario para facilitar el aprendizaje.

4. Implementación:

En esta fase el curso se pone en marcha y se pone a disposición de los estudiantes. Se configura la plataforma de enseñanza y se brinda capacitación a los instructores o facilitadores para garantizar el uso correcto de las herramientas.

Durante esta etapa se busca asegurar que los usuarios puedan acceder sin problemas a los contenidos. Se debe garantizar que la plataforma funcione de manera óptima y fácil.

5. Evaluación:

Se evalúa la efectividad del curso y su impacto en el aprendizaje de los estudiantes. Este proceso se lleva a cabo durante su desarrollo (evaluación formativa) y al finalizar (evaluación sumativa), para identificar áreas de mejora.

Este modelo facilita la creación de cursos en línea efectivos, asegurando que cada fase del proceso contribuya a una enseñanza de calidad. La evaluación juega un papel importante permitiendo ajustes en cada fase para mejorar la calidad y asegurar el logro de los objetivos.

2.1.3 Ventajas de ADDIE

Dentro de las ventajas que maneja el modelo ADDIE son:

- Enfoque estructurado: Facilita la planificación y ejecución del proyecto.
- Personalizable: Permite adaptarse según el público y los recursos disponibles.

- Efectivo y eficaz: Asegura una enseñanza de calidad alineada con los objetivos.
- Iterativo: Permite mejoras continuas.
- Integración tecnología: Integra herramientas digitales para crear cursos dinámicos y accesibles.

Estas ventajas hacen un modelo bien estructurado, capaz de adaptarse sin perder la efectividad para el desarrollo de recursos educativos digitales.

2.2 Modelo ASSURE

El modelo ASSURE es un sistema de diseño instruccional utilizado en la creación de software educativo, centrado en la integración de medios y tecnologías en el proceso de enseñanza-aprendizaje.

Fue desarrollado por Robert Heinich et al. (1999) como una metodología que permite mejorar la enseñanza mediante la incorporación efectiva de tecnologías educativas.

Su estructura permite adaptar los contenidos de enseñanza a las necesidades de los usuarios haciendo uso adecuado de herramientas tecnológicas.

2.2.1 Características de ASSURE

Entre los aspectos principales de este modelo destacan los siguientes aspectos:

- Variedad de Cursos: Ofrece una amplia gama de cursos.
- Integración de Tecnologías: Se centra en el uso de las tecnologías de manera efectiva para la mejora de la enseñanza.
- Flexibilidad: Capacidad de ser adaptado a diferentes contextos educativos.

2.2.2 Fases de ASSURE

El modelo ASSURE está compuesta por seis fases que guían el desarrollo del curso (Heinich y otros, 1999), como se observa en la Figura 2.6:

1. *Analyze* - Análisis de los estudiantes:
Fase inicial, donde se determina las características de los estudiantes, como su edad, estilos de aprendizaje, necesidades específicas, lo cual permite adaptar el contenido y los métodos de enseñanza adecuados.

2. **State** - Definición de Objetivos de Aprendizaje:
En esta etapa se establecen los objetivos de aprendizaje, que deben ser medibles y alineados con las necesidades de los estudiantes. Guían el diseño de la instrucción y aseguran que logren los resultados deseados.
3. **Select** - Selección de Métodos, Medios y Materiales:
Se seleccionan los métodos de enseñanza, los medios (videos, presentaciones), los materiales (libros, artículos) más adecuados para cumplir los objetivos de aprendizaje.
4. **Utilize** - Uso de Tecnologías y Materiales:
Se implementa los recursos seleccionados para el curso. Se verifica que los materiales funcionen correctamente en la plataforma LMS y sean accesibles para todos los usuarios. Además de capacitar a los docentes.
5. **Require** - Fomento de la Participación del Estudiante:
Se diseñan actividades que involucren de forma activa a los estudiantes, por ejemplo foros, ejercicios prácticos, actividades interactivas.
6. **Evaluate** - Evaluación y Revisión del Curso:
En esta fase se evalúa la efectividad del curso por medio de encuestas, análisis del desempeño de los estudiantes y retroalimentación de los participantes. Así se identifican áreas de mejora.

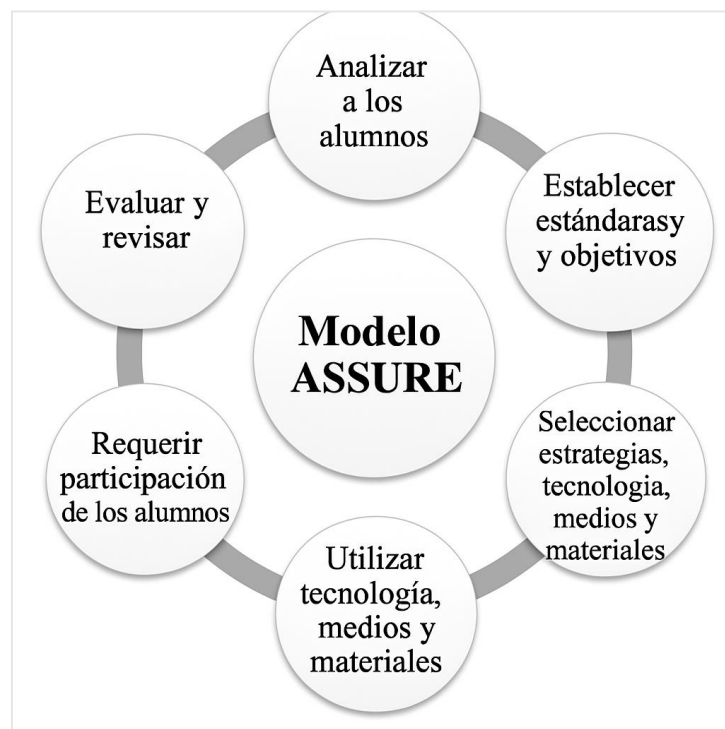


Figura 2.6. Fases del modelo ASSURE (Heinich y otros, 1999).

2.2.3 Ventajas de ASSURE

De las ventajas que presenta este modelo para desarrollar cursos en línea podemos encontrar las siguientes:

- Flexibilidad: Capacidad de aplicarse en diversos entornos educativos.
- Integración eficiente de tecnológicas: Facilita el uso de herramientas digitales, mejorando la interactividad.
- Participación de los estudiantes: Mejora la interacción de los estudiantes con el contenido promoviendo el aprendizaje.

2.3 Modelo SAM (*Successive Approximation Model*)

El modelo SAM fue desarrollado por Michael W. Allen, el cual permite realizar ajustes y mejoras tempranas y constantes durante todo el proceso de diseño, garantizando que el proyecto sea efectivo (Allen, 2012).

El modelo de aproximación sucesiva es un modelo de enfoque ágil e iterativo para el diseño instruccional centrado en la creación rápida de prototipos y retroalimentación inmediata durante todo el proceso.

2.3.1 Características de SAM

Algunas de las principales características que tiene este modelo son:

- Flexibilidad y adaptabilidad: Capacidad del curso para adaptarse rápidamente a los cambios y necesidades.
- Prototipado rápido: Capacidad de desarrollar de forma rápida los prototipos.
- Mejora continua: Promueve la mejora constante y la implementación de cambios.

2.3.2 Fases de SAM

El modelo Sam para el desarrollo de cursos tiene tres fases a seguir (Sánchez García y otros, 2016), tal como se muestran en la Figura 2.7.

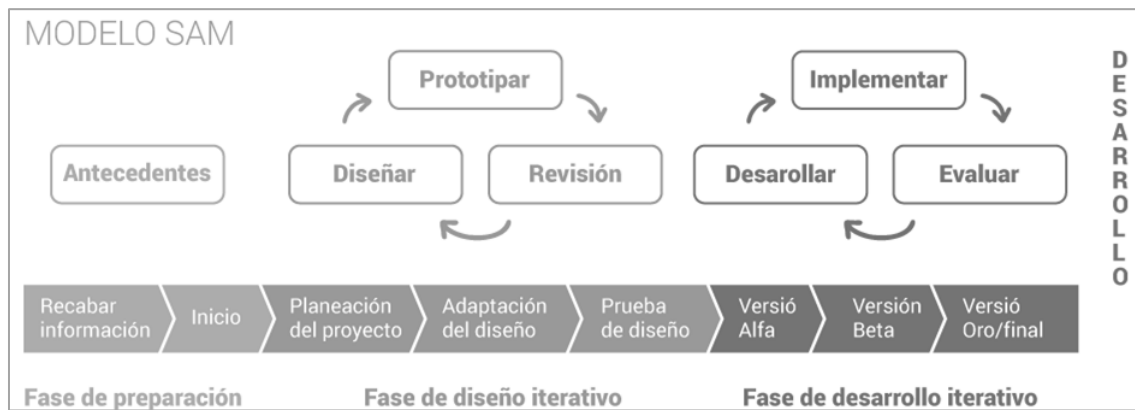


Figura 2.7. Fases de modelo SAM (Sánchez García y otros, 2016).

A continuación se describen cada una de las fases (Allen, 2012):

1. Preparación:

Esta fase es más rápida y dinámica. Incluye la práctica que fomenta la lluvia de ideas, creación de bocetos y prototipos iniciales.

Se involucra a todas las partes interesadas además, durante esta fase se recopila información y se inicia el proyecto.

2. Diseño iterativo:

Se enfoca en la planeación del proyecto y el desarrollo de proyectos. Se asignan roles a los miembros del equipo y establecen tiempos. Se hacen actividades del diseño y adaptación del diseño y se prueba el diseño.

3. Desarrollo iterativo:

En esta etapa se enfoca en la mejora y perfeccionamiento. Se crean diferentes versiones como: alfa, beta y final/oro.

Se evalúan y se ajustan continuamente para asegurar el cumplimiento de las necesidades del producto final.

2.3.4 Ventajas de SAM

Algunas de las ventajas que ofrece el modelo SAM son:

- **Retroalimentación constante:** Cuenta con ciclos constantes de revisión y retroalimentación desde las primeras etapas.
- **Interacción rápida:** Capacidad del desarrollo rápido de prototipos y la mejora continua.
- **Adaptación:** Permite el cambio de requisitos y cambios a lo largo del tiempo.

2.4 Modelo Gagné

El modelo de aprendizaje Gagné, desarrollado por Robert M. Gagné (1965), es un enfoque de diseño instruccional basado en el aprendizaje cognitivo. También conocido como “Las Nuevas Condiciones del Aprendizaje”, se centra en una serie de pasos que explican cómo las personas adquieren, procesan y retienen la información.

2.4.1 Características de Gagné

Entre las características principales de este modelo destacan los siguientes aspectos:

- Estructuras secuenciales: Define pasos claros y progresivos para garantizar el correcto aprendizaje.
- Flexibilidad: Aunque es secuencial, permite adaptaciones.
- Basado en el procesamiento de la información: Se enfoca en cómo el cerebro recibe, procesa y almacena la información.

2.4.2 Fases de Gagné

Gagné identifica cinco categorías principales de aprendizaje los cuales son: información verbal, habilidades intelectuales, estrategias cognitivas, habilidades motrices y actividades.

Las fases del aprendizaje según Gagné son, tal como se muestran en la Figura 2.8 (Universidad Católica Los Ángeles de Chimbote):

1. Motivación:
Se busca captar la atención de los alumnos y motivarlos. Se establecen los objetivos, muestra relevancia del contenido dentro de la vida y se busca despertar el interés
2. Comprensión o Aprehensión:
Se asegura que los usuarios comprendan la nueva información. Se muestra los recursos de manera clara, usando recursos visuales y ejemplos que faciliten la comprensión.
3. Adquisición:
Se busca ayudar a los alumnos a comprender nueva información. Se promueve la participación efectiva, se brinda oportunidades de prácticas guiadas.

4. Retención:
Se busca asegurar que se retenga la información a largo plazo mediante la utilización de resúmenes, repeticiones, relaciones y otros recursos faciliten la memorización
5. Recuperación de la Información:
Se facilita la recuperación de la información almacenada y promueve la práctica.
6. Generalización:
Se busca aplicar lo aprendido a diferentes situaciones donde se pueda aplicar el conocimiento.
7. Desempeño:
Se evalúa el aprendizaje y desempeño de los alumnos, mediante evaluaciones.
8. Retroalimentación:
Se brinda retroalimentación inmediata y específica después de las evaluaciones.

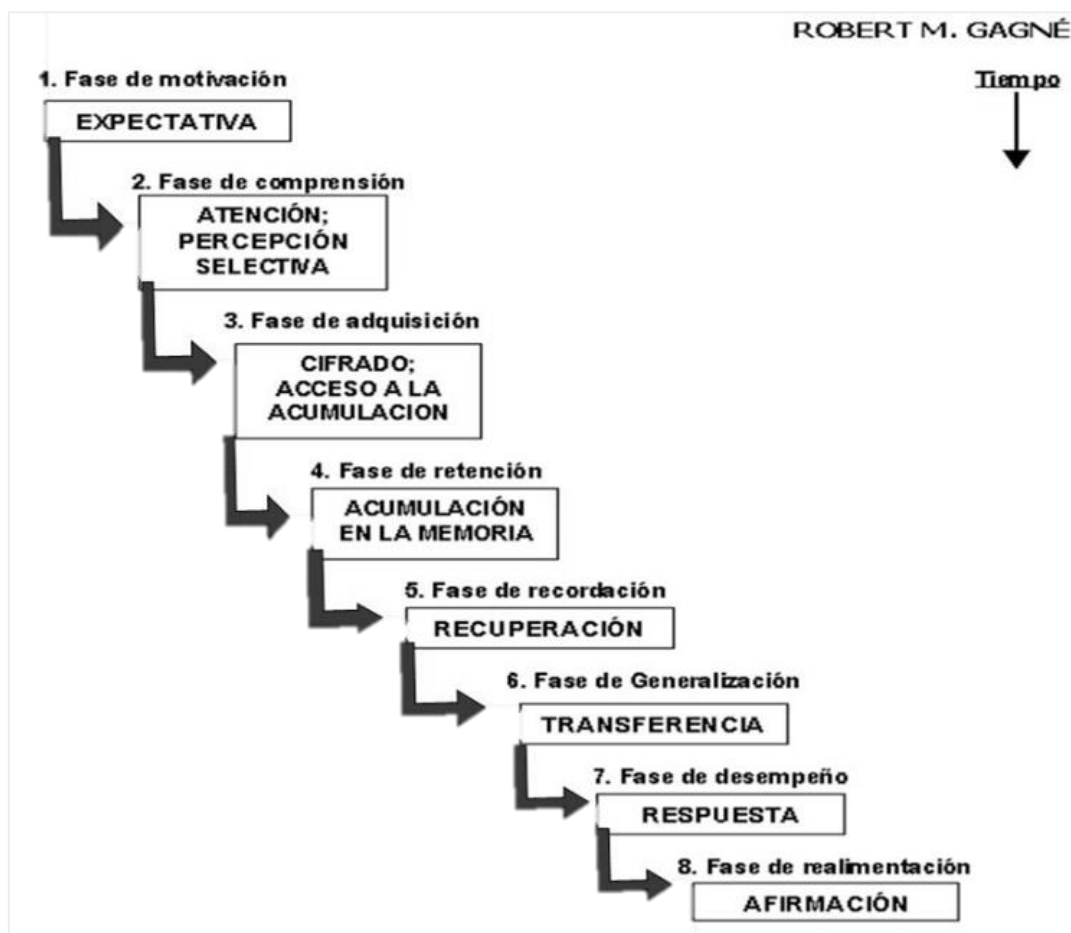


Figura 2.8. Modelo Gagné (Universidad Católica Los Ángeles de Chimbote, s.f.).

Gracias a estas fases, se puede desarrollar un curso que no solo transmita información, sino que también promueva el aprendizaje duradero.

Además, Gagné detalla nueve eventos de instrucción que facilitan el aprendizaje, junto a sus procesos cognitivos:

1. Conseguir la atención (recepción).
2. Informar a los estudiantes del objetivo (finalidad-utilidad).
3. Estimular la recuperación de aprendizajes previos (recuerdo).
4. Presentación de los estímulos (percepción selectiva).
5. Orientar el aprendizaje (codificación semántica).
6. Provocar el rendimiento (respuesta).
7. Proporcionar retroalimentación (refuerzo).
8. Evaluar el rendimiento (recuperación).
9. Mejorar la retención y transferencia (generalización).

2.4.3 Ventajas de Gagné

El modelo Gagné presenta muchas ventajas para el desarrollo de cursos entre ellos:

- Favorece la retención de conocimientos: Estimula el uso de conocimientos previos ayudando a retener información a largo plazo.
- Diferentes contextos: Se puede usar en la educación presencial, virtual y para formación empresarial y capacitación.
- Retroalimentación efectiva: Es clave la evaluación y corrección en cada etapa para mejorar el desempeño del usuario.

2.5 Comparación de modelos

Los modelos analizados para el desarrollo de cursos en línea comparten similitudes en sus objetivos generales y en su enfoque para la creación de contenidos educativos.

Sin embargo, cada modelo tiene características particulares en cuanto a sus fases de implementación, su nivel de adaptabilidad y su integración con distintas tecnologías.

La Tabla 2.3 muestra una comparación detallada de cuatro modelos para el desarrollo de cursos en línea: ADDIE, ASSURE, SAM y Gagné.

Tabla 2.3. Tabla comparativa de las metodologías para el desarrollo de cursos.

	ADDIE	ASSURE	SAM	Gagné
Enfoque	Secuencial e iterativo	Centrado en la integración de tecnología	Ágil e iterativo	Cognitivo, basado en el procesamiento de la información
Flexibilidad	Alta	Alta	Media	Media
Metodología de software	Tradicional, basada en un ciclo estructurado	Selección e integración de medios digitales	Ágil, adaptado a iteraciones rápidas	Basado en principios del aprendizaje humano
Compatibilidad con LMS	Alta	Alta	Media	Media
Fases que la integran	1.Análisis 2.Diseño 3.Desarrollo 4.Implementación 5.Evaluación	1.Análisis 2.Selección de Medios 3.Uso 4.Revisión 5.Evaluación 6.Integración	1.Diseño 2.Desarrollo 3.Evaluación rápida	1.Motivación 2.Comprensión 3.Adquisición 4.Retención 5.Recuperación 6.Generalización 7.Desempeño 8.Retroalimentación
Uso de SCORM	Sí	Sí	Depende del LMS	Sí
Facilidad de actualización	Media-Alta	Alta	Alta	Media
Accesibilidad	Alta	Alta	Media	Media
Medición de resultados	Alta	Alta	Alta	Alta
Adaptabilidad	Media	Alta	Alta	Media
Escalabilidad	Alta	Media	Alta	Media
Tiempo de desarrollo	Largo, debido a la planificación detallada	Medio, depende de la selección de medios	Corto, permite interacciones rápidas	Medio, con énfasis en la estructuración del contenido

Fuente: Autoría propia.

Esta comparación permite identificar las fortalezas y debilidades de cada modelo. Así, se facilita la elección del modelo más adecuado para llevar a cabo el desarrollo del curso SoftLearn, de acuerdo con sus necesidades.

2.6 Elección de la metodología a implementar

Después del análisis comparativo de los modelos instruccionales, se concluye que el modelo ADDIE es el más adecuado para el desarrollo del proyecto SoftLearn.

La elección del modelo ADDIE se basa en su estructura sistemática, su equilibrio pedagógico y lo tecnológico, y su compatibilidad con plataformas LMS y estándares educativos, como SCORM.

A continuación se mencionan las razones que justifican esta elección:

- Estructura organizada y control del proceso:
Al constar con cinco etapas, permite gestionar el curso de manera ordenada y controlada. Ya que se supervisa cada fase de desarrollo, asegurando el cumplimiento de los objetivos.
- Interactividad y aprendizaje activo:
Facilita la implementación de herramientas como foros, cuestionarios y evaluaciones, promoviendo la participación, ofreciendo contenido atractivo.
- Flexibilidad y adaptabilidad:
Permite la actualización continua de los contenidos del curso, al momento de identificar las áreas de mejora se pueden realizar ajustes sin afectar la estructura del curso.
- Compatibilidad con estándares educativos:
Facilita la integración de plataformas LMS y estándares como SCORM, asegurando la accesibilidad del curso desde diversos dispositivos.
- Equilibrio pedagógico y tecnológico:
Este modelo destaca por integrar eficazmente los aspectos pedagógicos con los tecnológicos, lo que resulta ideal para enseñar conceptos técnicos sobre el Análisis de Sistemas de Software, de manera clara, comprensible y accesible.
- Evaluación continua:
ADDIE contempla mecanismos de evaluación formativa y sumativa, lo que permite monitorear el impacto del curso en los estudiantes, detectar debilidades y realizar mejoras.

En resumen, ADDIE es una metodología que brinda un enfoque estructurado, flexible y centrado en la mejora continua, lo que lo convierte en la opción más conveniente para el desarrollo de SoftLearn. Además, su compatibilidad con plataformas LMS lo posiciona como la alternativa más adecuada para ofrecer una experiencia de aprendizaje efectiva y de calidad.

CAPÍTULO III. Estado del arte

En este capítulo se analiza una variedad de cursos orientados al aprendizaje del Análisis de Sistemas de Software, con el objetivo de facilitar a los estudiantes la práctica y el refuerzo de sus conocimientos. A continuación, se describen algunos de estos cursos.

3.1 Introducción a la Ingeniería del Software (UAMx - edX)

edX es una plataforma de cursos en línea, fundada en 2012 por el Instituto Tecnológico de Massachusetts (MIT) y la Universidad de Harvard. Su objetivo es dar acceso a la educación a todos mediante MOOCs por instituciones académicas.

La plataforma edX brinda a los estudiantes de todo el mundo acceder a contenido de alta calidad, ya sea sin costo o con opción de pago para obtener certificados oficiales. Su enfoque se basa en el aprendizaje autónomo, con recursos como videos, lecturas, foros y evaluaciones automatizadas (edX, 2023).

La Universidad Autónoma de Madrid, a través de la iniciativa UAMx, ofrece el curso “Introducción a la Ingeniería del Software” en la plataforma edX. Su propósito es brindar una visión completa sobre los métodos, técnicas y procedimientos fundamentales para llevar a cabo un proyecto de software con éxito, cumpliendo con los requisitos de tiempo, recursos y costos.

El programa aborda aspectos fundamentales de la Ingeniería del Software, y está organizado en módulos que integran diversos recursos didácticos como: videos explicativos, lecturas complementarias, foros de discusión y evaluaciones, como se observa en la Figura 3.9.

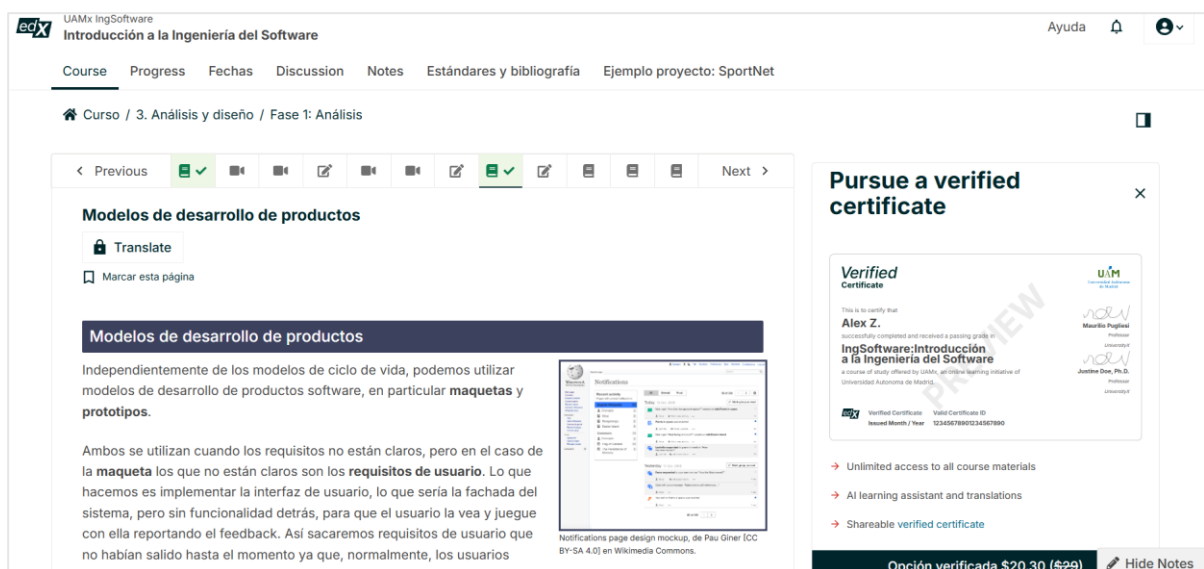


Figura 3.9. Interfaz del recurso “Introducción a la Ingeniería del Software” (edX, 2024).

3.2 Software Engineering (Quizlet)

Quizlet es una plataforma de aprendizaje en línea creada en 2005 por Andrew Sutherland, con el propósito de facilitar el estudio mediante el uso de tarjetas de memorización (*flashcards*), juegos interactivos, evaluaciones automatizadas y actividades colaborativas. Su principal objetivo es fomentar el aprendizaje autónomo y dinámico a través de la repetición espaciada y la gamificación (Quizlet, Inc., 2025).

En este contexto, se identifica un recurso en inglés titulado “*Software Engineering Study Guide*”, disponible en la plataforma, el cual está conformado por definiciones clave, conceptos fundamentales y descripciones breves sobre temas de Ingeniería de Software.

Está organizado como un conjunto de tarjetas digitales que abordan contenidos en Ingeniería de Software, que permiten a los usuarios repasar conceptos de manera rápida y visual, reforzando el aprendizaje mediante la práctica repetitiva.

También se incluyen ejercicios tipo fichas, aprender, probar, bloques, blast y combinar, que pueden ser útiles para la preparación de exámenes teóricos, como se observa en la Figura 3.10.

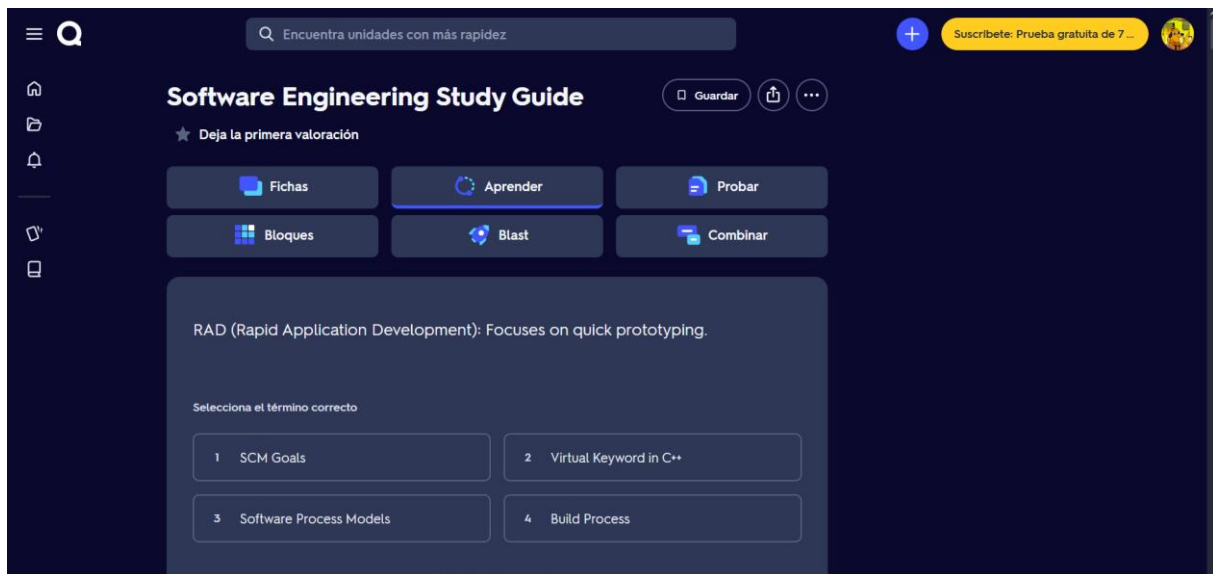


Figura 3.10. Interfaz del recurso “*Software Engineering Study Guide*” (Quizlet, s.f.).

Quizlet es útil para reforzar la memorización de conceptos fundamentales, si bien la plataforma no ofrece una estructura curricular integral ni actividades prácticas avanzadas. Además, carece de preguntas en escenarios reales, ya que solo manejar teoría, así se considera una herramienta complementaria en el proceso de aprendizaje.

3.3 Ingeniería de Software y Arquitectura de Software (Udemy)

Udemy es una plataforma educativa en línea fundada en 2010 por Eren Bali, Gagan Biyani y Oktay Caglar. Su objetivo general es permitir el acceso a cursos educativos accesibles para todo el mundo (Udemy, 2010).

A diferencia de otras plataformas tipo MOOC, Udemy permite que cualquier persona diseñe y publique sus propios cursos, estableciendo sus contenido, metodología y precios. Udemy se basa en un modelo de aprendizaje autónomo, flexible que permite a los estudiantes avanzar a su propio ritmo, acceder a los materiales de por vida.

En este contexto, se analiza el curso titulado “Ingeniería de Software y Arquitectura de Software”, disponible en español en la plataforma Udemy, como se observa en la Figura 3.11.

Este curso proporciona una introducción a los fundamentos de la Ingeniería de Software estructurado en 13 secciones con más de 50 clases en video pregrabados, materiales descargables, cuestionarios de autoevaluación, foros de discusión y certificados de finalización.

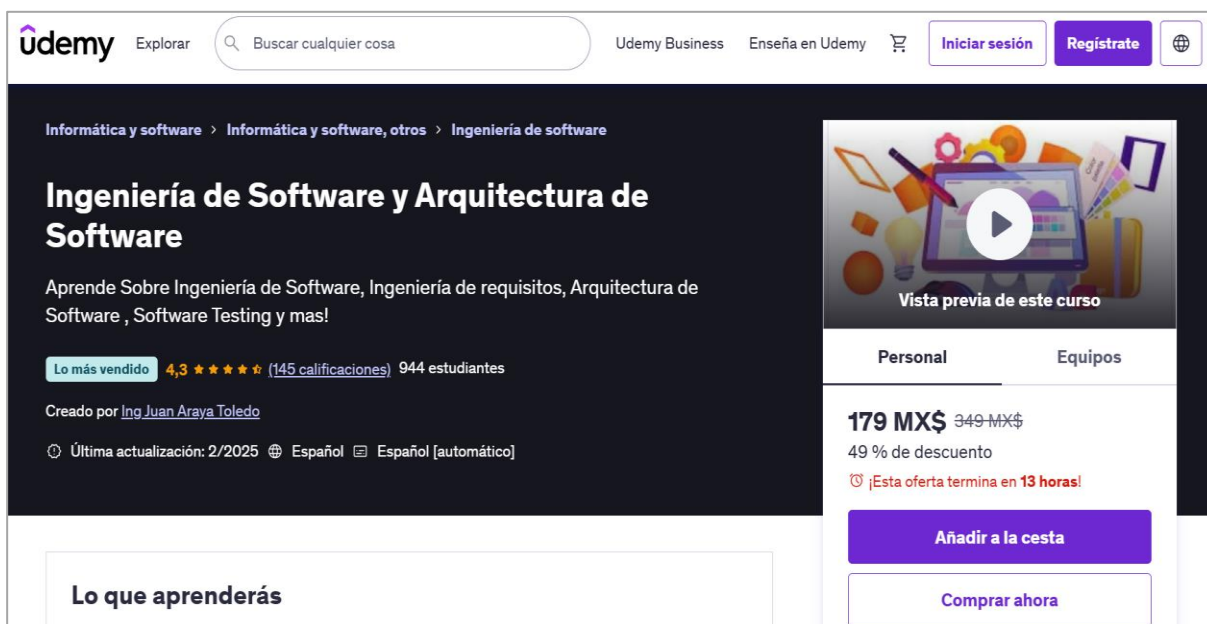
The image is a screenshot of the Udemy website interface for a course titled "Ingeniería de Software y Arquitectura de Software". The header includes the Udemy logo, a search bar, and navigation links like "Udemy Business" and "Enseña en Udemy". The course title is prominently displayed in a dark box with a play button icon and the text "Vista previa de este curso". Below the title, it says "Aprende Sobre Ingeniería de Software, Ingeniería de requisitos, Arquitectura de Software, Software Testing y mas!". The course is rated 4.3 stars with 145 reviews and has 944 students. It was created by Ing Juan Araya Toledo and was last updated in 2/2025. The price is shown as 179 MX\$ (49% discount from 349 MX\$) with a countdown timer indicating the offer ends in 13 hours. There are buttons for "Añadir a la cesta" and "Comprar ahora". A section titled "Lo que aprenderás" is partially visible at the bottom.

Figura 3.11. Interfaz del recurso “Ingeniería de Software y Arquitectura de Software” (Udemy, 2025).

La estructura de la plataforma lo hace adecuado tanto para estudiantes como para profesionales que buscan profundizar en estos conceptos.

Aunque Udemy ofrece un enfoque teórico-práctico presenta algunas restricciones, la resolución de ejercicios sin retroalimentación automática, además de que el acceso completo al curso requiere un pago, lo que puede representar una barrera para algunos usuarios.

3.4 Curso de Ingeniería de Software (NobleProg)

NobleProg es una plataforma de capacitación en línea y proveedor de formación profesional que ofrece cursos de alta calidad en diversas áreas, fundada en 2005, con la opción de formación tanto presenciales como en línea (NobleProg, 2025).

En caso del “Curso de Ingeniería de Software”, está diseñado para cubrir los aspectos fundamentales de la Ingeniería de Software, proporcionando tanto la teoría necesaria como la aplicación práctica.

La plataforma brinda acceso a recursos como videos educativos, ejercicios y evaluaciones, permitiendo a los estudiantes aplicar lo aprendido, como se observa en la Figura 3.12.

The screenshot displays the NobleProg website interface for the 'Curso de Ingeniería de Software'. The header features the NobleProg logo, 'Corporate Services', a search bar, and a navigation menu with links to 'Catálogo de Cursos', 'Gobierno', 'Consultoría', 'Recapacitación', 'Servicios', 'Nosotros', and a 'Contáctenos' button. The main content area has a breadcrumb trail: 'Inicio / Ingeniería De Software Capacitación / Curso De Ingeniería De Software'. The course title 'Curso de Ingeniería de Software' is prominently displayed, followed by a 5-star rating and '(2 Testimonios)'. A descriptive paragraph states: 'Software Engineering comienza donde se detiene la programación. En este curso aprenderás una base sólida en Software Engineering. Construye un puente entre la programación y las sólidas prácticas de ingeniería. El curso le brinda una descripción general de los conceptos básicos y el vocabulario que se usa típicamente en el campo. Durante el curso, aprenderás a utilizar muchas técnicas para planificar, implementar y validar proyectos de software y para evaluar y mejorar los procesos de desarrollo.' Below this, the 'Audiencia:' section notes: 'Este curso está dirigido a desarrolladores de software con diversos antecedentes que conocen al menos un lenguaje de programación.' and a 'Leer menos' link. The 'Programa del Curso' section lists topics for 'Día 1: Proceso' (Introducción, ¿Qué es la Ingeniería de Sistemas?, ¿Qué es Software Engineering?, ¿Por qué Software Engineering es difícil?, Requisitos del sistema, Modelado de sistemas), 'Software Engineering Procesos' (¿Qué es un proceso de desarrollo de software?, Procesos Tradicionales, Agile Procesos, Procesos para construir Sistemas de Alta Integridad, Normas y sistemas de certificación), 'Sistemas de alta integridad' (¿Qué son los sistemas de alta integridad?, El papel de la cultura de gestión de calidad, Razones de los errores, Estándar de sistemas críticos, Ingeniería de confiabilidad), and 'Mantenimiento de software'. On the right, a sidebar shows '35 Horas', a 'CONSULTA DEL CURSO' button, a 'CONSULTAS' section with a 'Número de participantes' set to 1, radio buttons for 'En línea' (selected) and 'Aula', a 'Reservar - \$ 4344' button, and 'Precio por Participante'. A note at the bottom right states: 'Los cursos de formación abiertos requieren más de 5 participantes.' and a small icon indicates 'La ayuda no está en línea...'.

Figura 3.12. Interfaz del recurso “Curso de Ingeniería de Software” (NobleProg, 2025).

Aunque NobleProg cuenta con aspectos prácticos de la Ingeniería de Software, algunas reseñas mencionan la limitación en la cantidad de ejercicios prácticos, así mismo este es un curso de paga y para tomarlo de forma presencial está disponible únicamente en Estados Unidos.

3.5 Análisis comparativo de herramientas de aprendizaje

En la Tabla 3.4 se ofrece un análisis comparativo del curso de SoftLearn en relación con otros cursos en línea, destacando las características y ventajas de cada uno.

Tabla 3.4. Tabla comparativa de cursos de aprendizaje en línea.

Plataforma	Introducción a la Ingeniería del Software (UAMx - edX)	Software Engineering (Quizlet)	Ingeniería de Software y Arquitectura de Software (Udemy)	Curso de Ingeniería de Software (NobleProg)	SoftLearn (propuesta)
Costo	Pago	Gratuito	Pago	Pago	Gratuito
Enfoque	Introducción a la Ingeniería de Software	Reforzamiento de conceptos básicos	Desarrollo de Software y Principios de Arquitectura	Aplicación práctica de Ingeniería de Software	Análisis de Sistemas Software
Interactividad	Baja	Alta	Media	Alta	Alta
Retroalimentación	Lenta	Inmediata	Lenta	Moderada	Inmediata
Tipo de contenido	Teoría, videos, lecturas, exámenes	Tarjetas de memoria, definiciones, conceptos clave	Lecciones en video, cuestionarios, pruebas	Lecciones teóricas y prácticas, análisis de casos	Casos prácticos, ejercicios interactivos, teoría
Modalidad	En línea, autónomo	En línea, autónomo	En línea, autónomo	En línea, presencial o en vivo	En línea, autónomo
Aplicación práctica	Baja	Media	Alta	Alta	Alta

Fuente: Autoría propia.

SoftLearn se destaca entre otras plataformas por ser una herramienta completamente gratuita, especializada en la enseñanza de Análisis de Sistemas de Software, permitiendo a los estudiantes un aprendizaje más práctico mediante actividades interactivas, ejercicios de caso de uso y retroalimentación inmediata, lo que facilita la aplicación efectiva de sus conocimientos.

A diferencia de otras plataformas que se enfocan en aspectos teóricos o tienen una interactividad limitada. Además, la mayoría de las plataformas que abordan la Ingeniería de Software suelen tener costos elevados, lo que limita la accesibilidad para muchos estudiantes.

CAPÍTULO IV. Desarrollo del proyecto SoftLearn

Este capítulo describe el proceso de desarrollo de SoftLearn, un curso en línea de tipo SPOC diseñado específicamente para la enseñanza del Análisis de Sistemas de Software en el área de Ingeniería de Software.

El nombre de SoftLearn se deriva de la combinación de palabras “*Soft*” que hace referencia a software, y “*Learn*”, que significa aprender en inglés. Este nombre refleja el propósito del proyecto.

Para garantizar la calidad pedagógica y tecnológica del curso, se adopta el modelo ADDIE, que ofrece un enfoque estructurado y flexible para el diseño y desarrollo de la enseñanza. Este modelo asegura que los aspectos tecnológicos como pedagógicos se aborden de manera efectiva y eficiente, ofreciendo un curso de calidad.

A continuación, se detallan las fases de implementación del modelo ADDIE, explicando cómo cada una de ellas se aplica específicamente al curso SoftLearn.

4.1 Análisis

Dentro de esta fase se definen las bases para el desarrollo de SoftLearn, definiendo los objetivos, recursos necesarios, público objetivo y el cronograma de trabajo.

Esta etapa es clave para garantizar que el curso esté alineado con las necesidades de los estudiantes y asegurar el éxito del proyecto.

4.1.1 Definición de objetivos

Los objetivos generales y específicos del curso SoftLearn se definen de la siguiente manera:

- Objetivo general:

Desarrollar un SPOC orientado al fortalecimiento del aprendizaje en el Análisis de Sistemas de Software, utilizando la metodología ADDIE proporcionando contenidos educativos, actividades y evaluaciones automáticas que preparen a los estudiantes y refuercen sus conocimientos, mediante programas de código abierto como Chamilo y HotPotatoes.

- **Objetivos específicos:**
 - Facilitar la comprensión de conceptos clave en el Análisis de Sistemas de Software, a través de actividades interactivas que refuercen el aprendizaje.
 - Apoyar la preparación de los estudiantes en la materia de Ingeniería de Software, desarrollando módulos que cubran los temas clave del programa, permitiendo a los estudiantes practicar y consolidar sus conocimientos.
 - Proporcionar retroalimentación en tiempo real, que permita a los estudiantes identificar sus errores y corregirlos inmediatamente, mejorando su comprensión de los temas abordados.
 - Fomentar el aprendizaje autónomo, ofreciendo acceso al aprendizaje ubicuo.

4.1.2 Análisis del público objetivo

SoftLearn está diseñado para abordar las necesidades de diferentes grupos de estudiantes, específicamente aquellos que están involucrados en el estudio de Análisis de Sistemas de Software.

El público objetivo de este proyecto está compuesto por:

- Estudiantes de la Licenciatura en Ciencias Computacionales (LCC) en tercer semestre que cursan la materia de Ingeniería de Software.
- Estudiantes en formación interesados en fortalecer su comprensión sobre el Análisis de Sistemas de Software.
- Estudiantes que necesitan reforzar conceptos para presentar un examen extraordinario en la materia de Ingeniería de Software.

4.1.3 Análisis pedagógico

En esta fase, se definen los contenidos pedagógicos que se incluirán en SoftLearn, ayudando a que los estudiantes refirman u obtengan los conocimientos necesarios.

El curso se organiza en módulos estructurados con los siguientes temas clave dentro de la disciplina del Análisis de Sistemas de Software:

1. Tipos de Requerimientos.
2. Técnicas y herramientas para la obtención, análisis, priorización y validación de requerimientos.
3. Técnicas y herramientas de documentación de requerimientos.

A través de la enseñanza de estos temas, los alumnos desarrollarán competencias y habilidades como:

- Competencia en el análisis de requerimientos.
- Habilidad en la recolección de requerimientos.
- Competencia en la documentación de requisitos.

4.1.4 Análisis técnico

Para garantizar la correcta implementación y funcionamiento de SoftLearn, se seleccionan las herramientas tecnológicas más adecuadas para asegurar su eficiencia, accesibilidad y escalabilidad.

Este análisis técnico abarca desde la elección de la plataforma LMS hasta la infraestructura de hardware y software usadas para su desarrollo e implementación.

- Plataforma y desarrollo:

SoftLearn se desarrollará en Chamilo, un sistema de gestión de aprendizaje (LMS) de código abierto que permitía administrar el contenido educativo, gestionar usuarios y estructurar módulos de aprendizaje de manera eficiente.

Su interfaz intuitiva y flexibilidad lo hacen la opción ideal para la enseñanza en línea, ya que permite la integración de materiales multimedia, cuestionarios interactivos y foros de discusión.

Para apoyar la experiencia de aprendizaje, se utilizará HotPotatoes, una herramienta especializada en la creación de ejercicios dinámicos como crucigramas, cuestionarios de opción múltiple, ejercicios de rellenar espacios en blanco y actividades de emparejamiento.

Además, para complementar el contenido visual del curso, se utilizará Canva, una plataforma de diseño gráfico en línea y de este modo facilitar la creación de materiales educativos atractivos y de este modo reforzar o complementar los temas abordados en SoftLearn.

- Infraestructura tecnológica:

Para un correcto funcionamiento y disponibilidad del curso estará alojado en un servidor en la nube adquirido a través de HostGator, asegurando la escalabilidad, accesibilidad y seguridad.

El alojamiento en la nube facilita el mantenimiento y actualización del sistema sin interrupciones.

4.1.5 Cronograma de trabajo

Como se observa en la Figura 4.13 mediante un diagrama de Gantt, las fechas de inicio y finalización de cada fase del desarrollo de SoftLearn, desde el análisis hasta la validación:



Figura 4.13. Diagrama Gantt sobre el desarrollo del curso.

4.2 Diseño

El diseño es una fase clave en el desarrollo de SoftLearn, ya que permite estructurar el proyecto utilizando toda la información obtenida anteriormente para diseñar cada una de las partes del curso.

Durante esta fase, se definen la arquitectura pedagógica, la interfaz de usuario y los elementos interactivos que integran el curso, asegurando que la plataforma sea intuitiva, accesible y efectiva para el aprendizaje.

El diseño se enfoca en dar una experiencia de usuario intuitiva, así los estudiantes pueden aprender a su propio ritmo, mediante lecciones modulares, actividades interactivas y retroalimentación en tiempo real.

4.2.1 Diseño pedagógico

El diseño pedagógico de SoftLearn se enfoca en brindar un aprendizaje activo, autonomía y evaluación continua.

- El curso se divide en módulos, cada uno con un conjunto de lecciones teóricas, ejercicios interactivos y evaluaciones formativas.
- Los módulos están diseñados para que los estudiantes avancen de lo básico a lo complejo.
- Cada módulo incluye autoevaluaciones interactivas y un examen final.

4.2.2 Diseño de la interfaz de usuario

La interfaz de SoftLearn está basada en la estructura base ofrecida por la plataforma Chamilo, asegurando así una experiencia de usuario intuitiva y eficiente, como se observa en la Figura 4.14.

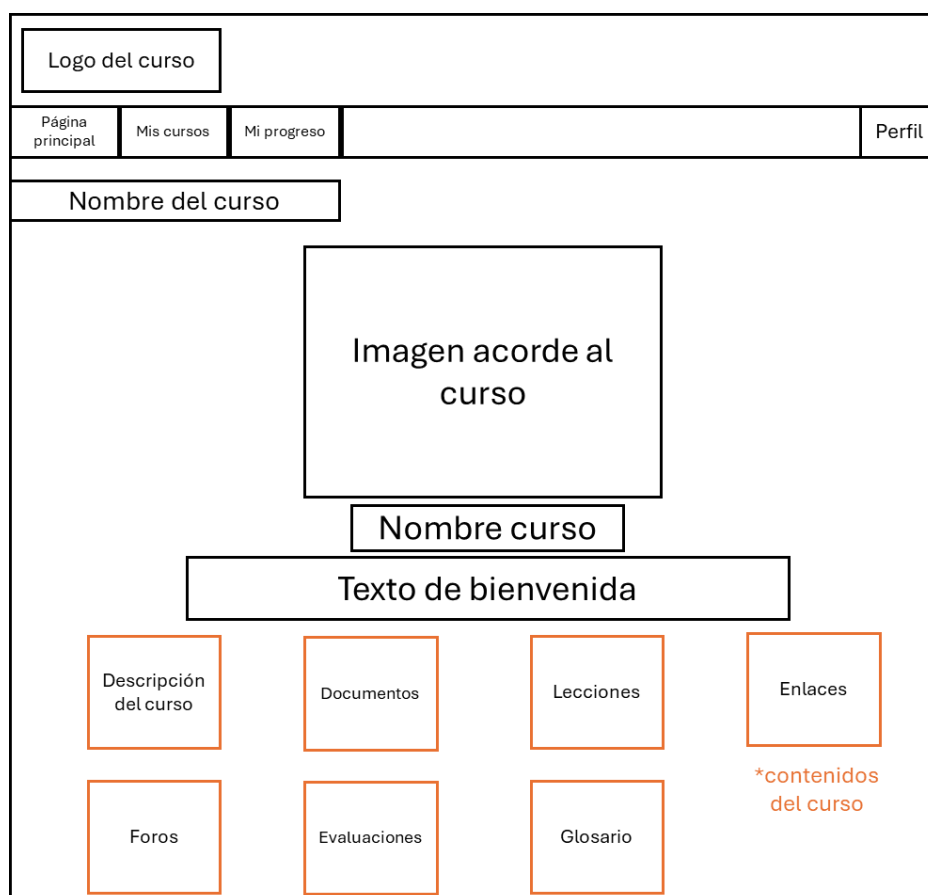


Figura 4.14. Bosquejo de interfaz del curso en Chamilo. Fuente: Autoría propia.

Chamilo es un LMS que brinda una interfaz preconfigurada que simplifica la navegación y permite un acceso eficiente a los recursos educativos. Como se observa en la Figura 4.14 el bosquejo del diseño base de la interfaz de un curso nuevo en Chamilo.

Sin embargo, el administrador de SoftLearn tiene la capacidad de personalizar por completo la interfaz, configurando tanto las funcionalidades disponibles como los aspectos visuales, adaptándose mejor a los objetivos del proyecto.

Esto abarca desde las herramientas y funcionalidades que serán accesibles para los estudiantes, eliminando aquellas que no sean imprescindibles para el curso. Así se mejora la plataforma creando un entorno ordenado y enfocándose en los recursos y actividades más relevantes para el aprendizaje.

Además, se realizan cambios visuales en la interfaz, sustituyendo los logotipos y elementos gráficos originales de Chamilo por los de SoftLearn, lo que ofrece la personalización y adaptación de la plataforma, modificando directamente el código fuente.

Por lo tanto, se asegura que la plataforma cuente con una navegación fluida desde cualquier dispositivo con acceso a internet.

4.2.3 Funcionamiento y elementos de SoftLearn

Los elementos que componen a SoftLearn han sido diseñados con el objetivo de garantizar que los estudiantes puedan interactuar de manera efectiva con el curso. Cada componente cumple una función específica en la creación del curso, brindando una experiencia completa.

A continuación, se describen los componentes y como estos se integran en la plataforma SoftLearn:

- Portada de inicio del curso:
La primera pantalla que los estudiantes encuentran al ingresar al curso es la portada, donde se les da la bienvenida y un breve texto introductorio del curso.

Además, se presentan las opciones que tienen durante el curso.

- Descripción del curso:
En el apartado “Descripción de curso” se brinda información clave sobre los siguientes aspectos:

- Descripción general:
Se proporciona una breve y concisa descripción del proyecto SoftLearn, explicando en qué consiste la plataforma y cuál es su enfoque principal.
 - Objetivos de Aprendizaje:
Se detallan los objetivos del curso de manera precisa, indicando claramente los resultados que se espera que los estudiantes alcance al completar el curso.
 - Temario:
El temario incluye los temas que se abordarán a lo largo del curso. Estos temas cubren aspectos clave, brindando a los estudiantes una visión completa del contenido que reforzarán.
- La taxonomía del curso:
En la Figura 14.5 se muestra la estructura de los módulos y lecciones en SoftLearn.

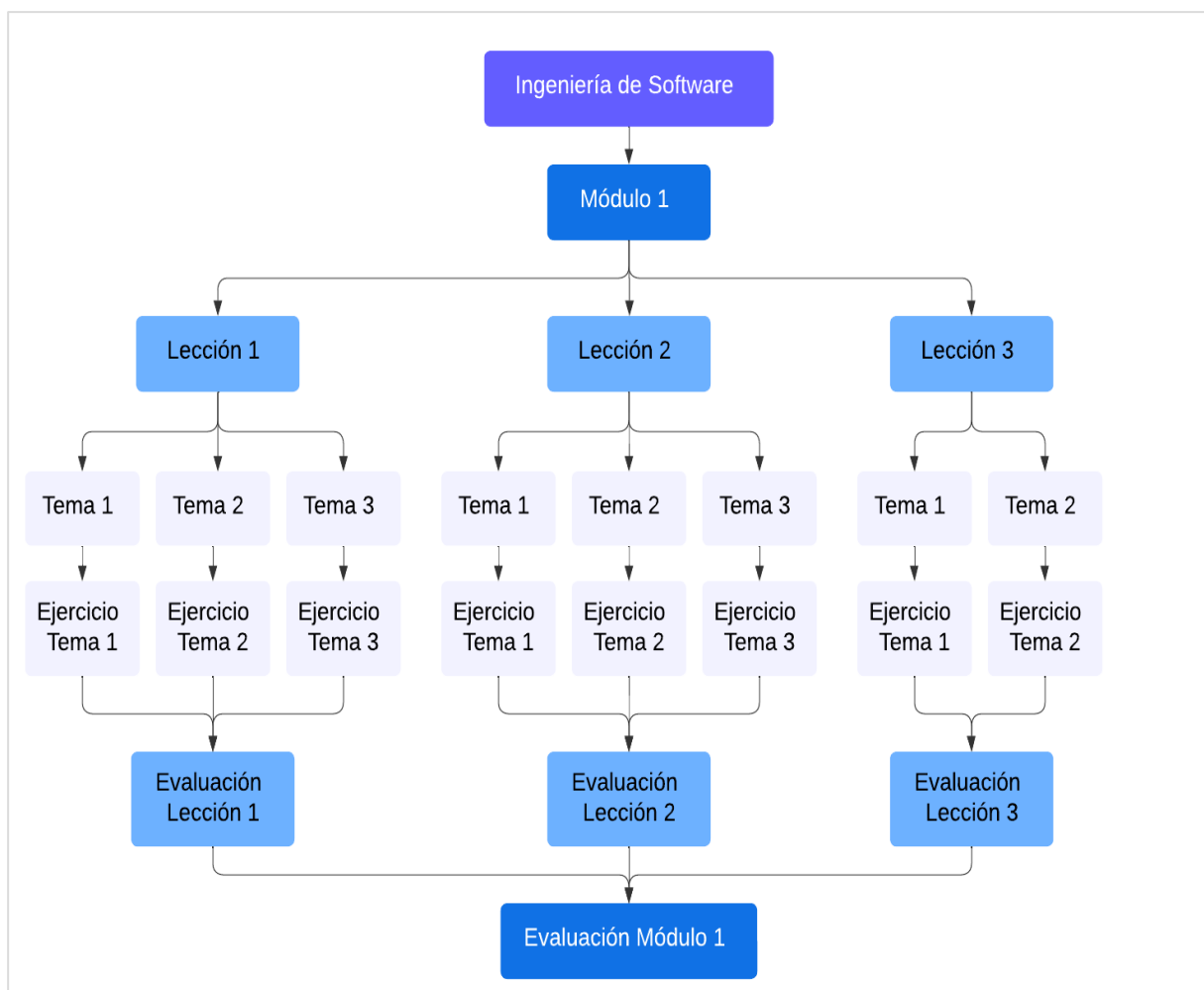


Figura 4.15. Taxonomía del curso SoftLearn. Fuente: Autoría propia.

- Contenido/Lecciones:

El contenido educativo incluye temas clave de la Ingeniería de Software, como el Análisis de Sistemas de Software, la gestión de requerimientos y la documentación de estos.

Además del contenido teórico contará con elementos visuales para captar la atención de los estudiantes y sean capaces de asimilar la información de una manera más dinámica y fácil.

- Ejercicios/Actividades:

Al finalizar cada lección del curso se realizará un ejercicio o serie de ejercicios interactivos que refuercen los conceptos aprendidos. Entre los cuales se incluirá preguntas de teoría, análisis de casos y la resolución de problemas relacionados con la Ingeniería de Software.

Los ejercicios ofrecen una retroalimentación en tiempo real a los estudiantes y de este modo puedan identificar sus fallas y puntos de mejora. Además de asegurar que los estudiantes cuenten con los conocimientos necesarios para continuar al siguiente tema.

- Evaluación:

Al finalizar el módulo se incluye una evaluación que mide el progreso de los estudiantes. Esta evaluación será un conjunto de ejercicios interactivos que los estudiantes deberán responder, las cubrirán los puntos clave del módulo.

Los diferentes tipos de ejercicios utilizados como se mencionan más adelante se aplicarán tanto en los ejercicios y actividades al final de cada lección como en la evaluación final del módulo.

No se seguirá un orden o tema específico para el uso de los tipos de ejercicios, ya que serán usados de manera flexible para evaluar los conocimientos y habilidades.

- Recursos adicionales:

Además del contenido educativo, SoftLearn cuenta con recursos adicionales que complementan el aprendizaje:

- Glosario:

El glosario proporciona definiciones claras de los términos técnicos de Ingeniería de Software.

Esto permite a los estudiantes consultar y comprender un concepto que no les sea familiar, fortaleciendo su vocabulario técnico y mejorando su capacidad de interpretar textos.

- Documentos:
La sección de documentos incluye materiales adicionales como libros en formato pdf.

- Foro:
El foro proporciona un espacio interactivo para que los estudiantes puedan compartir dudas, experiencias y comentarios entre ellos.

Este recurso fomenta el aprendizaje colaborativo, ya que tanto como estudiantes como profesores pueden colaborar y de este modo resolver preguntas de manera colectiva.

- Enlaces:
Los enlaces están basados en las recomendaciones de la guía EGEL sobre la Ingeniería de Software, específicamente en el área del Análisis de Sistemas de Software.

Brindando una base sólida y confiable para que los estudiantes puedan investigar más a fondo los temas abordados.

4.2.3.1 Actividades interactivas en SoftLearn

SoftLearn utiliza herramientas interactivas integradas en Chamilo y HotPotatoes para ofrecer una variedad de ejercicios dinámicos que faciliten tanto el aprendizaje como la evaluación de los estudiantes.

- **Ejercicios interactivos en Chamilo:**

Chamilo como LMS, ofrece una gran variedad de herramientas que permiten crear ejercicios interactivos dentro de la plataforma. Además, permite activar la opción de retroalimentación inmediata, donde los estudiantes recibirán de manera automática una evaluación detallada sobre sus respuestas al terminar el ejercicio.

Los tipos de ejercicios interactivos disponibles incluyen:

- Arrastrable:
Permite a los estudiantes mover elementos y colocarlos en los lugares correctos. Es útil para organizar secuencias o relacionar conceptos, facilitando conexiones visuales e interactivas.
- Respuesta única:
Presenta varias opciones, pero solo una es correcta. Sirve para evaluar conocimientos específicos de manera precisa.

- Respuesta múltiple:
Ofrece varias opciones de respuesta, donde puede haber una o más correctas. Es útil para evaluar conocimientos concretos y la capacidad de discernir entre múltiples posibilidades.
- Relacionar:
Se deben conectar elementos de dos listas, como términos con definiciones o conceptos con ejemplos. Ayuda a fortalecer las conexiones entre conceptos y mejorar la comprensión.
- Rellenar espacios:
Se completan frases escribiendo las palabras faltantes. Es útil para evaluar vocabulario y conocimiento técnico.

Estos ejercicios están diseñados para apoyar la comprensión teórica y fomentar la práctica activa de los estudiantes.

- Ejercicios interactivos con HotPotatoes:

HotPotatoes es una herramienta importante para la creación de actividades interactivas en SoftLearn.

Este software permite la creación de diferentes ejercicios interactivos que ayudan a completar el contenido y pueden ser exportadas en formato SCORM e integrarse a Chamilo, lo que permite aprovechar los beneficios de ambas plataformas.

HotPotatoes permite brindar una retroalimentación inmediata a los estudiantes. Cada ejercicio se puede configurar para ofrecer comentarios inmediatos después de responder una actividad, además de poder incluir explicaciones adicionales para reforzar el aprendizaje.

Algunos de los ejercicios que ofrece HotPotatoes son:

- JQuiz:
Permite crear cuestionarios con opción múltiple, respuestas cortas y verdadero/falso. Es útil para evaluar conocimientos de manera estructurada y variada.
- JCloze:
Facilita crear ejercicios donde los estudiantes deben completar las frases con la palabra correcta. Es ideal para reforzar vocabulario y comprensión de conceptos clave.

- JMatch:
Desarrolla actividades donde se deben relacionar conceptos con sus definiciones o características. Es útil para reforzar el conocimiento teórico de manera visual e interactiva.
- JMix:
Permite crear ejercicios en los que los estudiantes deben organizar palabras o frases para formar secuencias coherentes. Es útil para identificar etapas en un proceso y mejorar la comprensión estructural de la información
- JCross:
Genera crucigramas interactivos que ayudan a los estudiantes a aprender vocabularios técnicos de forma práctica.

Cada una de estas herramientas fomenta el aprendizaje interactivo y la retención de conocimientos de manera efectiva.

4.3 Desarrollo

La fase de desarrollo de SoftLearn está enfocada en la creación del contenido educativo y las actividades interactivas que permitirán a los estudiantes reforzar sus conocimientos en Análisis de Sistemas de Software.

A continuación, se detallan los elementos clave que conforman esta fase.

4.3.1 Contenido educativo

El contenido educativo de SoftLearn ha sido diseñado para ofrecer una experiencia de aprendizaje completa y dinámica, combinando lecciones teóricas, actividades prácticas y evaluaciones interactivas.

A continuación, se describen los principales elementos del curso y como se implementaron en SoftLearn:

- Portada del curso:
La Figura 4.16 muestra la portada, la cual está conformada por un mensaje de bienvenida, además de las secciones que el estudiante puede explorar.

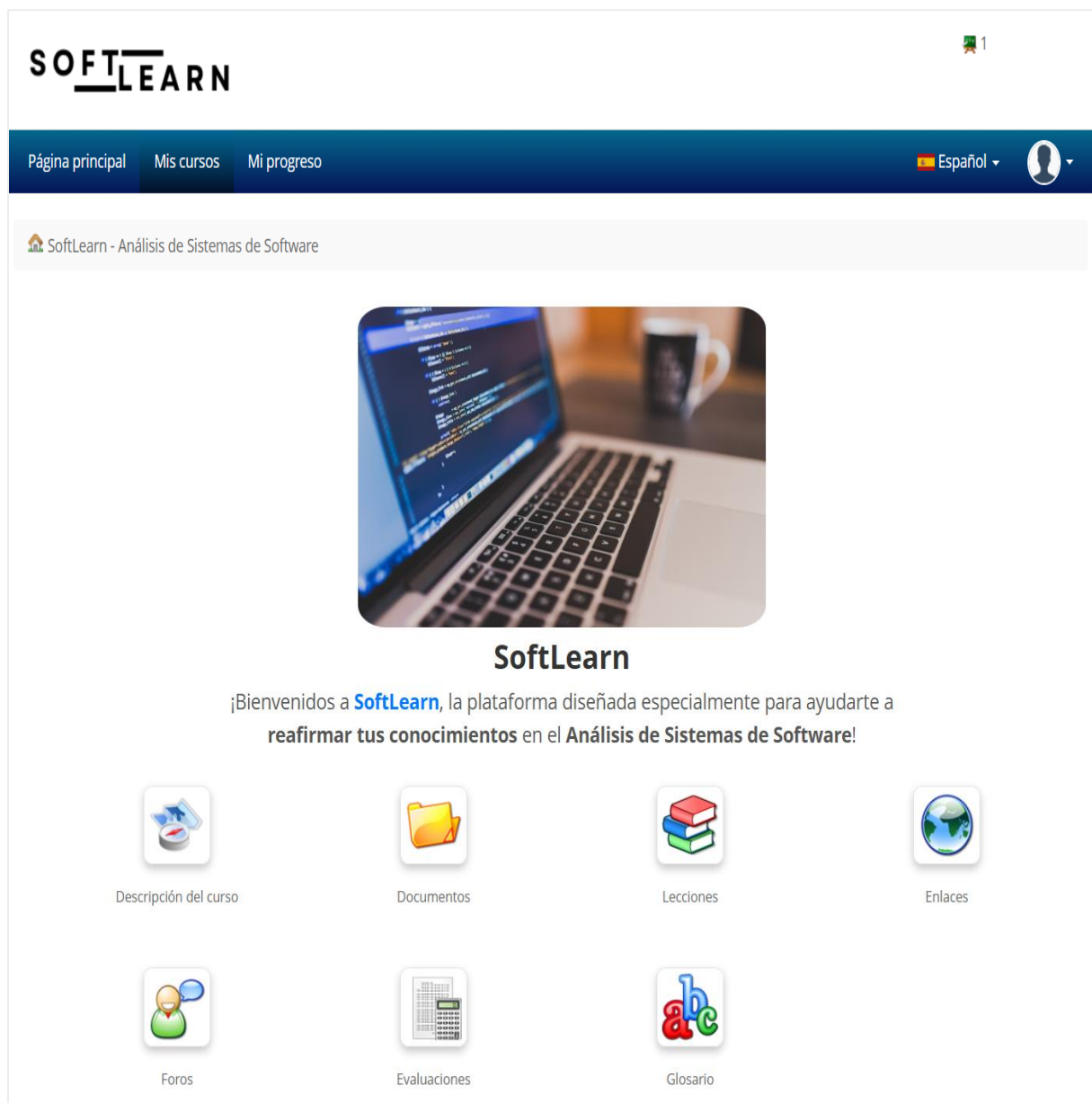


Figura 4.16. Portada de la interfaz SoftLearn.

- **Descripción del curso:**

El curso define con claridad lo que el usuario va a aprender, por lo que su descripción del curso se centra en establecer expectativas claras sobre los temas cubiertos y los objetivos están bien planteados y delimitados, puesto que son una parte fundamental del curso.

Además de la descripción general, el curso incluye un temario detallado que incluye toda la información necesaria sobre los temas que abarca el mismo en módulos progresivos.

Tal como se observa en la Figura 4.17, el curso proporciona un acceso organizado a elementos facilitando la navegación.




Página principal Mis cursos Mi progreso

 Español
 

 SoftLearn - Análisis de Sistemas de Software / Descripción del curso / Descripción del curso

Descripción del curso

El curso **SoftLearn** está diseñado para estudiantes que desean reforzar y ampliar sus conocimientos en los temas clave de la Ingeniería de Software, con un enfoque especial en el Análisis de Sistemas de Software. Está orientado tanto a quienes buscan fortalecer su comprensión para aprobar la materia de Ingeniería de Software como a aquellos interesados en prepararse para exámenes extraordinarios o evaluaciones específicas dentro de este campo.

Este curso interactivo combina teoría y práctica para ofrecer una experiencia de aprendizaje dinámica y efectiva. A través de módulos temáticos, actividades interactivas y ejercicios prácticos, los estudiantes podrán desarrollar competencias esenciales en áreas como el análisis de sistemas, la gestión de requerimientos y la documentación de proyectos de software.

Objetivos

▪ **Objetivo General:**

Desarrollar una plataforma educativa interactiva, diseñada para reforzar los conocimientos en Ingeniería de Software, con un enfoque particular en el Análisis de Sistemas de Software. La plataforma proporcionará contenidos educativos, actividades interactivas y evaluaciones, apoyando tanto a los estudiantes en su preparación para la materia de Ingeniería de Software como a aquellos que necesiten cursar un examen extraordinario o realizar una evaluación relacionada.

▪ **Objetivos Específicos:**

- Facilitar la comprensión de conceptos clave en Ingeniería de Software, especialmente en el Análisis de Sistemas de Software, mediante actividades interactivas que refuercen el aprendizaje y fomenten la práctica autónoma.
- Apoyar la preparación de los estudiantes para la materia de Ingeniería de Software y/o exámenes extraordinarios, desarrollando módulos que cubran los temas relevantes, permitiendo a los estudiantes practicar y consolidar sus conocimientos.
- Proporcionar retroalimentación en tiempo real, permitiendo que los estudiantes identifiquen y corrijan sus errores inmediatamente, mejorando su comprensión de los temas abordados y optimizando su desempeño académico.

Temario

1 Análisis de Sistemas de Software

1.1 Tipos de Requerimientos

- Clasificación de los requerimientos
- Características de los requerimientos
- ISO 9126

1.2 Técnicas y herramientas para la obtención, análisis, priorización y validación de requerimientos

- ¿Qué es Ingeniería de Requerimientos?
- Procesos de requerimientos
- Técnicas y herramientas de elicitación de requerimientos

1.3 Técnicas y herramientas de documentación de requerimientos

- Modelado de sistemas
- Documentación de los requerimientos

Figura 4.17. Interfaz Descripción del curso.

- **Taxonomía del curso:**
Tal como se visualiza en la Figura 4.18 se observa la estructura jerárquica del contenido, organizada en el módulo de Análisis de Sistemas de Software, con lecciones, actividades y evaluaciones:

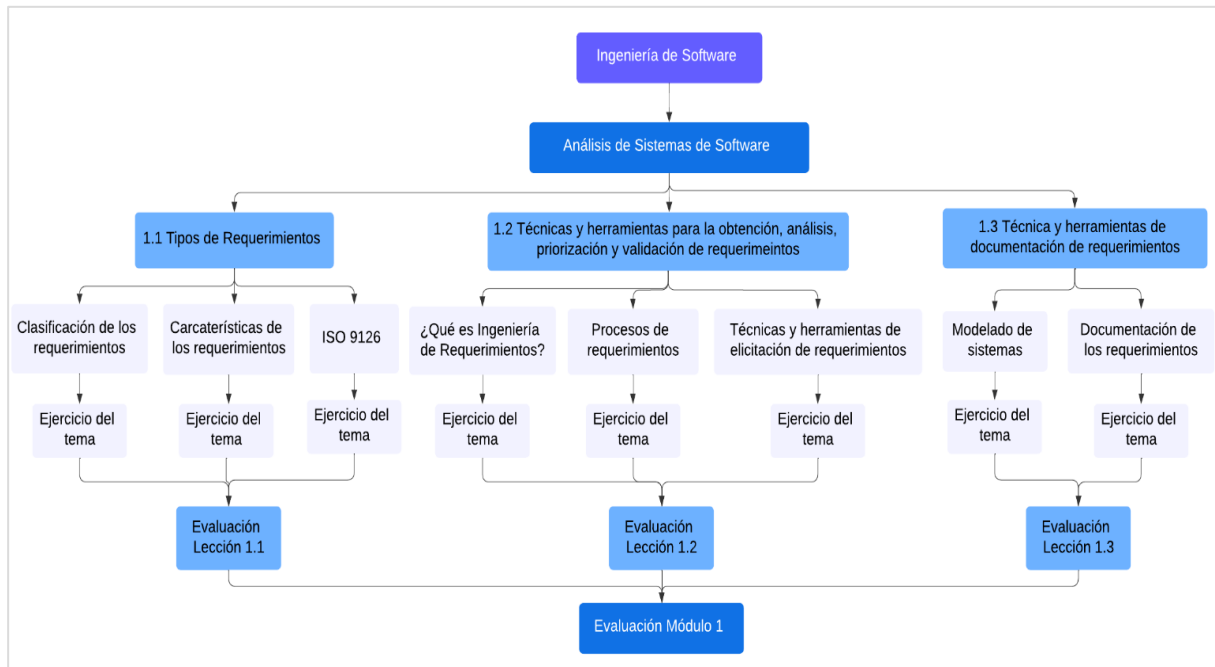


Figura 4.18. Taxonomía del curso con lecciones, ejercicios y la evaluación.

- **Lecciones:**
Como se observa en la Figura 4.19, se desarrollan los temas según el temario presentado anteriormente.

Página principal

Mis cursos

Mi progreso

🇪🇸 Español

👤

🏠

SoftLearn - Desarrollo de Software / Lecciones

1. Análisis de Sistemas de Software

Título	Progreso	Acciones
1.1. Tipos de requerimientos	40%	
1.2. Técnicas y herramientas para la obtención, análisis, priorización y validación de requerimientos	0%	
1.3. Técnicas y herramientas de documentación de requerimientos	0%	

Figura 4.19. Interfaz del Módulo 1: Análisis de Sistemas de Software.

Al momento de ingresar a las lecciones, se incluye un menú del lado izquierdo para acceder a cada uno de los temas y actividades que lo integran tal como se ilustra en la Figura 4.20.

Clasificación de Requerimientos por Dominio

Debido a la diversidad de los requerimientos, Sommerville (2005), sugiere organizarlos en tres dominios:

- Requerimientos del usuario
- Requerimientos del sistema
- Especificaciones del diseño de software

1. Requerimientos de Usuario

Son declaraciones en lenguaje natural (o en diagramas informales) de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar. Están enfocados en lo que el usuario espera del sistema.

- Ejemplos:
 - Los usuarios deben poder iniciar sesión utilizando su dirección de correo electrónico y contraseña.
 - El sistema debe permitir a los usuarios cargar archivos de hasta 5 MB.

2. Requerimientos del Sistema

Establecen con detalle las funciones, servicios y restricciones operativas del sistema. Deben definir exactamente lo

Figura 4.20. Interfaz de la Lección 1.1: Tipos de requerimientos.

▪ Evaluación:

Como se observa en la Figura 4.21, se evaluará en el Módulo 1: Análisis de Sistemas de Software.

Tipo	Nombre	Descripción	Ponderación	Resultado	Clasificación	Mejor calificación	Promedio	Acciones
	Lección 1.1		20	0 / 20	-	0 % (0 / 20)	-	
	1.1. Tipos de requerimientos Lecciones		-	33 % (33 / 100)	1 / 1	44 % (44 / 100)	39 % (39 / 100)	-
	Ejercicios Lección 1.1 Ejercicios		20	-				
	Lección 1.2		20	0 / 20	-	0 % (0 / 20)	-	
	1.2. Técnicas y herramientas para la obtención, análisis, priorización y validación de requerimientos Lecciones		-	-				
	Ejercicios Lección 1.2 Ejercicios		20	-				
	Lección 1.3		20	0 / 20	-	0 % (0 / 20)	-	
	1.3. Técnicas y herramientas de documentación de requerimientos Lecciones		-	-				
	Ejercicios Lección 1.3 Ejercicios		20	-				
	Módulo Análisis de Sistemas de Software		40	0 / 40	-	0 % (0 / 40)	-	
	Evaluación Análisis de Sistemas de Software Ejercicios		40	-				
-	Total	-	100	0 / 100	-	0 / 100	0 / 0	

Figura 4.21. Evaluación del Módulo 1: Análisis de Sistemas de Software.

4.3.2 Ejercicios interactivos en SoftLearn

Las actividades interactivas son de gran importancia en el entorno de aprendizaje de SoftLearn, ya que guían a los estudiantes a través del proceso de aprendizaje, proporcionándoles retroalimentación inmediata, reforzando los conocimientos adquiridos de forma efectiva.

Dentro de SoftLearn, los ejercicios interactivos se implementan mediante diversas herramientas educativas. Entre las plataformas utilizadas se encuentran Chamilo y HotPotatoes.

A continuación, se presentan algunos ejemplos de cómo se han aplicado al curso:

Ejercicios interactivos de Chamilo

A continuación, se presentan algunos ejemplos de los ejercicios implementados en el curso SoftLearn de acuerdo con los diferentes tipos de ejercicios que conforman la plataforma de Chamilo.

- Respuesta única:
Como se muestra en la Figura 4.22, el ejercicio de tipo “Respuesta Única” plantea una pregunta sobre la clasificación de requerimientos funcionales.

Modificar pregunta: Respuesta única

Pregunta

¿Cuál de los siguientes es un requerimiento funcional?

≡ Parámetros avanzados

Respuestas

Nº	Verdadero	Respuesta	Comentarios	Puntuación
1	(x)	El sistema debe permitir a los usuarios crear, modificar y eliminar cuentas.	Correcto. Este es un requerimiento funcional porque especifica una acción que el sistema debe realizar.	10
2	()	El sistema debe tener una disponibilidad del 99.9%.	Incorrecto. Esto es un requerimiento no funcional de fiabilidad.	
3	()	El sistema debe cumplir con la normativa GDPR.	Incorrecto. Cumplir con el GDPR es un requerimiento externo relacionado con normativas.	

Figura 4.22. Ejemplo de ejercicio de “Respuesta única”.

- **Respuesta múltiple:**
Como se muestra en la Figura 4.23, el ejercicio de tipo “Respuesta Múltiple” plantea una pregunta sobre la clasificación de requerimientos no funcionales.

Modificar pregunta: Respuesta multiple

Pregunta ¿Cuáles de las siguientes opciones describen correctamente un requerimiento no funcional?

Respuestas

Nº	Verdadero	Respuesta	Comentarios	Puntuación
1	☒	El sistema debe responder a las solicitudes de los usuarios en menos de 2 segundos.	Correcto. El tiempo de respuesta es un aspecto de rendimiento, que es un requerimiento no funcional.	5
2	☐	El sistema debe permitir a los usuarios modificar su perfil.	Incorrecto. Permitir modificar el perfil es una funcionalidad específica y corresponde a un requerimiento funcional.	
3	☒	El sistema debe ser accesible en dispositivos móviles.	Correcto. La accesibilidad en dispositivos móviles es una característica de portabilidad, que es un requerimiento no funcional.	5
4	☐	El sistema debe generar reportes de ventas mensuales.	Incorrecto. La generación de reportes es una función específica que el sistema debe realizar, por lo tanto, es un requerimiento funcional.	

Figura 4.23. Ejemplo de ejercicio de “Respuesta múltiple”.

- **Rellenar blanco:**
Como se muestra en la Figura 4.24, el ejercicio de tipo de “Rellenar blanco” requiere que los estudiantes completen las oraciones con los términos correctos relacionados con los diferentes tipos de requerimientos.

Modificar pregunta: Rellenar blancos

Pregunta Completa las siguientes oraciones con los términos correctos relacionados con los tipos de requerimientos.

Escriba el texto debajo, y use corchetes [...] para definir uno o más espacios en blanco



- Los Requerimientos [funcionales, Funcionales] describen las funciones específicas que el sistema debe realizar, como crear cuentas de usuario o generar reportes.
- Los Requerimientos [no funcionales, No Funcionales, no Funcionales] se enfocan en las características del sistema, como rendimiento, usabilidad y seguridad, pero no en funciones específicas.
- Los Requerimientos [externos, Externos] provienen de normativas o regulaciones externas que el sistema debe cumplir, como el cumplimiento de la ley GDPR.
- Los Requerimientos [organizacionales, Organizacionales] están relacionados con las políticas y procedimientos internos de la organización que afectan el desarrollo del sistema.
- Los Requerimientos [de usuario, de Usuario] reflejan las necesidades y expectativas que los usuarios tienen sobre el sistema, describiendo cómo esperan que funcione y qué servicios debe proporcionar.

Figura 4.24. Ejemplo de ejercicio de “Rellenar blancos”.

- **Relacionar:**
Como se muestra en la Figura 4.25, el ejercicio de tipo “Relacionar” consiste en emparejar diferentes tipos de requerimientos con sus ejemplos correspondientes.

Modificar pregunta: Relacionar con combinación exacta

Pregunta Empareja el tipo de requerimiento con su ejemplo.

≡ Parámetros avanzados

Relacionar

Nº	Respuesta	Corresponde a
1	Requerimiento de usuario	B
2	Requerimiento externo	A
3	Requerimiento de producto	C

Nº	Respuesta
A	El sistema debe cumplir con la normativa GDPR.
B	El usuario debe poder iniciar sesión con una contraseña.
C	El sistema debe ser fácil de usar para los usuarios finales.

Puntuación 10

Figura 4.25. Ejemplo de ejercicio de “Relacionar”.

- **Arrastrable:**
Como se muestra en la Figura 4.26, el ejercicio de tipo de “Arrastrable” permite a los estudiantes relacionar cada tipo de requerimiento con su categoría correspondiente mediante la función de arrastrar y soltar.

Modificar pregunta: Coincidencia arrastrable

Pregunta Arrastra cada tipo de requerimiento a la categoría que mejor lo describa.

≡ Parámetros avanzados

Relacionar

Nº	Respuesta	Corresponde a	Puntuación
1	Requerimientos Funcionales	A	5
2	Requerimientos No Funcionales	B	5

Nº	Respuesta
A	El sistema debe permitir a los usuarios modificar y eliminar sus cuentas.
B	El sistema debe responder en menos de 2 segundos.

Figura 4.26. Ejemplo de ejercicio de “Coincidencia arrastrable”.

Ejercicios interactivos en HotPotatoes:

A continuación, se muestran algunos ejemplos de los diferentes tipos de ejercicios creados con HotPotatoes que se implementaron en SoftLearn.

- JQuiz (cuestionario de preguntas):

La Figura 4.27 muestra un ejercicio JQuiz, donde se plantea una pregunta sobre la clasificación de requerimientos en un sistema de gestión de biblioteca.

Ejercicio Clasificación de los Requerimientos

Quiz

1 / 3 => [Mostrar todas las preguntas](#)

Buscar y Reservar un Libro en el Sistema de Gestión de Bibliotecas:

En el sistema de gestión de bibliotecas, los usuarios pueden buscar libros y realizar reservas. El sistema debe cumplir con ciertos requerimientos funcionales y no funcionales para garantizar que los usuarios puedan encontrar rápidamente los libros disponibles y reservarlos de manera segura.

¿Cuál de los siguientes es un ejemplo de requerimiento funcional para el caso de "Buscar y Reservar un Libro"?

- A. ☐ El sistema debe responder a las consultas de búsqueda en menos de 2 segundos.
- B. ☐ El sistema debe permitir a los usuarios buscar libros por título, autor o ISBN.
- C. ☐ El sistema debe cumplir con las normativas de protección de datos personales.
- D. ☐ El sistema debe ser fácil de usar para personas con conocimientos básicos de informática.

Figura 27. Ejemplo de ejercicio tipo JQuiz.

- JMatch (emparejamiento):

Como se observa en la Figura 4.28, el ejercicio de JMatch permite a los estudiantes emparejar clave con sus definiciones dentro del estándar ISO 9126.

Ejercicio ISO 9126

Ejercicio de emparejamiento

Relaciona los elementos de la derecha con los elementos de la izquierda.

[Verificar](#)

Funcionalidad	???
Fiabilidad	???
Usabilidad	???
Eficiencia	???
Mantenibilidad	???
Portabilidad	???

[Verificar](#)

Figura 28. Ejemplo de ejercicio tipo JMatch.

- o JCross (crucigramas):

Como se muestra en la Figura 4.29, el ejercicio de tipo JCross consiste en completar un crucigrama con términos clave relacionados con las técnicas de obtención de requerimientos.

Técnicas de Obtención de Requerimientos

Complete el crucigrama, luego haga clic en "Verificar" para verificar su respuesta.
Si está atascado, puede hacer clic en "Pista" para obtener una letra gratis.
Haga clic en un número en la cuadrícula para ver la pista o pistas para ese número. No inserte espacios entre palabras, ya que se cuentan como caracteres.

Horizontal:

1. Técnica que agrupa usuarios, procesos o componentes para determinar los límites del sistema y las interfaces entre ellos.
2. Método de recopilación de ideas en grupo donde se suspenden los juicios críticos inicialmente para fomentar la creatividad.
4. Técnica en la que se elaboran cuestionarios para recopilar información estructurada de muchos usuarios.
5. Técnica que clasifica y organiza ideas en grupos similares durante una sesión de lluvia de ideas o grupo nominal.
6. Técnica utilizada para detallar la interacción entre actores del sistema y el sistema mismo, a menudo representada con diagramas de casos de uso.
7. Técnica que permite a los usuarios ver una simulación del sistema para dar retroalimentación sobre el diseño y la funcionalidad.
8. Técnica de obtención que se basa en observar a los usuarios en su entorno de trabajo para comprender sus necesidades reales.
9. Técnica en la que un facilitador hace preguntas detalladas para obtener información de un entrevistado sobre los requerimientos del sistema.

Vertical:

1. Técnica de análisis que implica revisar documentos existentes para obtener requerimientos de sistemas actuales o previos.
3. Técnica para examinar sistemas existentes con el objetivo de identificar funciones que puedan ser útiles en el nuevo sistema.

Verificar

Figura 4.29. Ejemplo de ejercicio tipo JCross.

- JCloze (completar espacios en blanco):
Como se ilustra en la Figura 4.30, el ejercicio de JCloze consiste en completar frases con los términos correctos sobre características de los requerimientos.

Ejercicio Características de los Requerimientos
Ejercicio de relleno de huecos

Rellena todos los huecos y pulsa "Comprobar" para comprobar tus respuestas. Use el botón "Pista" para obtener una carta gratis si una respuesta le está dando problemas. También puedes hacer clic en el botón "[?]" para obtener una pista. ¡Ten en cuenta que perderás puntos si pides pistas o pistas!

Instrucción: Completa los espacios en blanco con la palabra correcta para describir cada característica de un buen requerimiento.

Un requerimiento debe estar [?] por escrito, proporcionando una base clara para la [?] y evitando malentendidos.

El requerimiento debe ser [?] es decir, debe ser posible de [?] o verificar para confirmar si se ha cumplido correctamente.

Un buen requerimiento debe ser [?] con una redacción clara y directa, evitando [?] o explicaciones innecesarias.

Figura 4.30. Ejemplo de ejercicio tipo JCloze.

- JMix (ordenación):
Como se muestra en la Figura 4.31, el ejercicio de JMix consiste en ordenar fragmentos de texto para formar una oración coherente sobre la importancia de los requerimientos.

Ejercicio Importancia de los requerimientos
Ejercicio de palabras mezcladas

Ordena las partes para formar una oración. Cuando crea que su respuesta es correcta, haga clic en "Verificar" para verificar su respuesta. Si te quedas atascado, haz clic en "Pista" para encontrar la siguiente parte correcta.

del cliente.

ya que ayudan

a planificar,

y asegurar que

Los requerimientos son esenciales para el éxito del software,

reducir riesgos

el producto cumpla

con las necesidades

Figura 4.31. Ejemplo de Ejercicio tipo JMix.

Con las actividades de ambas herramientas se crean evaluaciones para calificar los conocimientos del usuario, con la finalidad de ver el avance que tuvo el usuario durante el curso.

4.3.3 Recursos adicionales

SoftLearn incluye recursos adicionales para reforzar el aprendizaje de los estudiantes. A continuación se describen los recursos que se encuentran en SoftLearn:

- **Glosario:**
Se incluyen términos que pueden ser complicados para los usuarios, como se muestra en la Figura 4.32. Funciona como guía para enriquecer su vocabulario en conceptos técnicos de Ingeniería de Software, además de permitir descargarlo.



The screenshot shows the SoftLearn interface for the 'Glosario' (Glossary) section. The header includes the SoftLearn logo and navigation links: 'Página principal', 'Mis cursos', and 'Mi progreso'. The language is set to 'Español'. The breadcrumb trail is 'SoftLearn - Análisis de Sistemas de Software / Glosario / Lista'. Below the breadcrumb is a search bar with a 'Buscar' button. A table displays the following data:

Término ↓	Definición
Benchmarking	Proceso de comparación entre un sistema o proceso con estándares o competidores, con el fin de evaluar y mejorar el rendimiento.
Branch	En el contexto de diagramas de secuencia, "branches" se utiliza para modelar decisiones y caminos alternativos en el flujo de actividades del sistema.
Ciclo de Vida del Desarrollo de Software (SDLC)	Proceso estructurado que incluye fases como planificación, análisis, diseño, implementación y mantenimiento.
Diagrama	Los diagramas, especialmente los de flujo, de contexto y de casos de uso, son herramientas esenciales en el análisis de sistemas para modelar y entender los procesos y estructuras del sistema.

Figura 4.32. Interfaz del apartado de "Glosario".

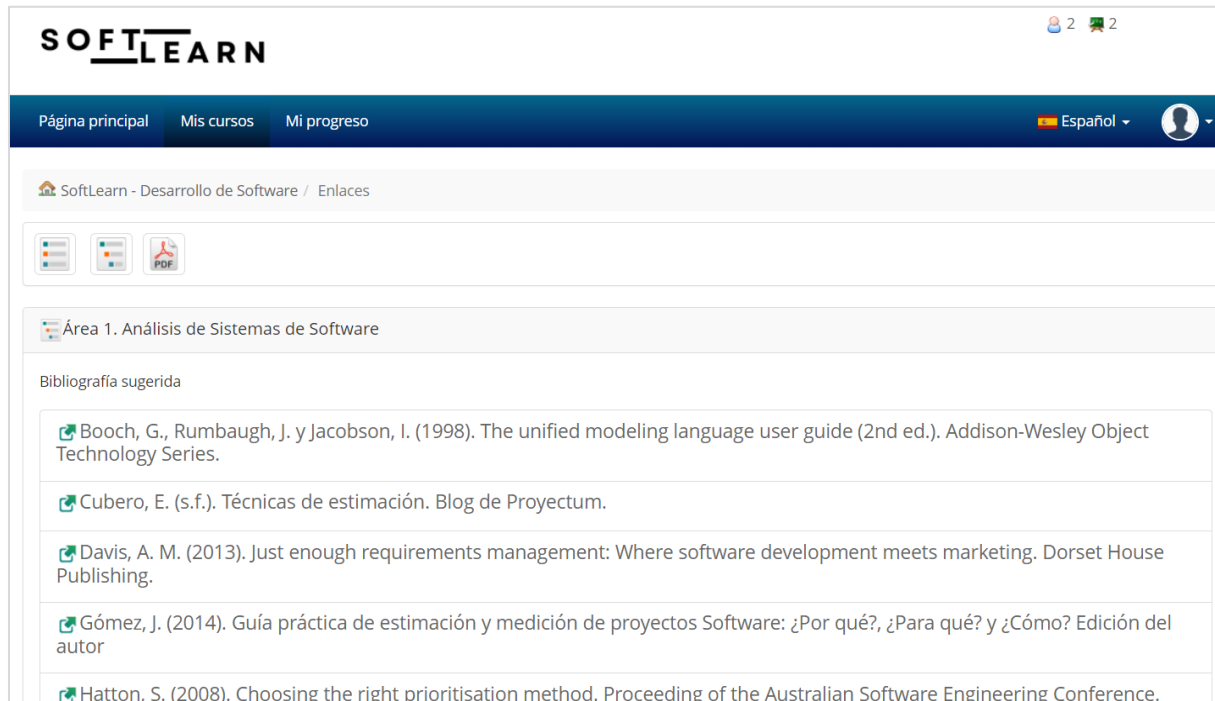
- **Foro:**
En esta sección brinda un espacio abierto para que los estudiantes puedan hacer preguntas y comentarios entre ellos, como se observa en la Figura 4.33.



The screenshot shows the SoftLearn interface for the 'Foros' (Forums) section. The header is identical to the previous figure. The breadcrumb trail is 'SoftLearn - Desarrollo de Software / Categorías de foro / Foros'. Below the breadcrumb is a search bar. The main section is titled 'Dudas y Preguntas sobre el Tema'. It features a 'Preguntas y Respuestas' (Questions and Answers) section with a welcome message: 'Bienvenidos al foro de preguntas y respuestas. Este es un espacio abierto para que puedas hacer preguntas y compartir tus dudas. Otros estudiantes o el instructor podrán responderte y ayudarte a aclarar conceptos. Antes de publicar, revisa si tu pregunta ya ha sido respondida. ¡También puedes contribuir ayudando a tus compañeros con sus dudas!'. There is a small icon of a person and a speech bubble next to the text.

Figura 4.33. Interfaz principal del apartado de "Foros".

- Documentos y enlaces:
Ambas secciones permiten acceder a los documentos y a los enlaces de referencia, como se observa en las Figuras 4.34 los enlaces y en la Figura 4.35 los archivos pdf.



SOFTLEARN

Página principal Mis cursos Mi progreso Español

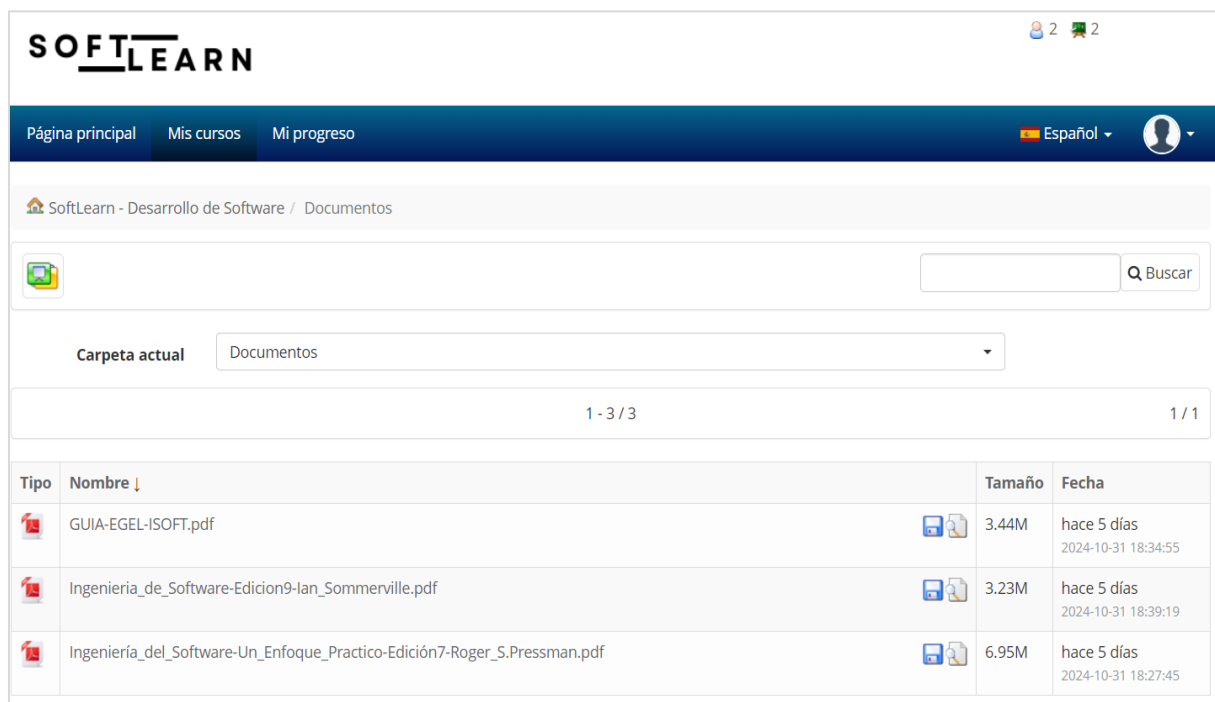
SoftLearn - Desarrollo de Software / Enlaces

Área 1. Análisis de Sistemas de Software

Bibliografía sugerida

- Booch, G., Rumbaugh, J. y Jacobson, I. (1998). The unified modeling language user guide (2nd ed.). Addison-Wesley Object Technology Series.
- Cubero, E. (s.f.). Técnicas de estimación. Blog de Proyectum.
- Davis, A. M. (2013). Just enough requirements management: Where software development meets marketing. Dorset House Publishing.
- Gómez, J. (2014). Guía práctica de estimación y medición de proyectos Software: ¿Por qué?, ¿Para qué? y ¿Cómo? Edición del autor
- Hatton, S. (2008). Choosing the right prioritisation method. Proceeding of the Australian Software Engineering Conference.

Figura 4.34. Interfaz del apartado de “Enlaces”.



SOFTLEARN

Página principal Mis cursos Mi progreso Español

SoftLearn - Desarrollo de Software / Documentos

Carpeta actual Documentos

1 - 3 / 3 1 / 1

Tipo	Nombre ↓	Tamaño	Fecha
	GUIA-EGEL-ISOFIT.pdf	3.44M	hace 5 días 2024-10-31 18:34:55
	Ingenieria_de_Software-Edicion9-Ian_Sommerville.pdf	3.23M	hace 5 días 2024-10-31 18:39:19
	Ingeniería_del_Software-Un_Enfoque_Practico-Edición7-Roger_S.Pressman.pdf	6.95M	hace 5 días 2024-10-31 18:27:45

Figura 4.35. Interfaz del apartado de “Documentos”.

4.4 Implementación

La fase de Implementación se centra en la puesta en funcionamiento de SoftLearn, asegurando que el curso esté completamente operando y listo para ser utilizado por los estudiantes. Esta etapa incluye el despliegue del curso en el LMS.

4.4.1 Despliegue de la plataforma y configuración del curso

El curso SoftLearn fue desplegado al principio en un entorno de forma local y después se desplegó en línea, alojado en un servidor de HostGator con un dominio propio, lo que garantiza su escalabilidad, seguridad y disponibilidad las 24 horas.

A continuación, se describe el procedimiento realizado en la instalación:

1. Desarrollo e instalación local de Chamilo:

Para el desarrollo del curso se configuró el entorno de trabajo en XAMPP. El proceso se describe a continuación:

- Se instala XAMPP en el equipo desde el sitio web oficial (<https://www.apachefriends.org/>).
- Después de terminar la instalación en la ruta `C:\xampp` se encuentra una carpeta de almacenamiento que pertenece al servidor local de XAMPP.
- Dentro de dicha carpeta se encuentra otra llamada “htdocs” y en esta se guarda el instalador de Chamilo su última versión, disponible en la página oficial (<https://chamilo.org/es/descargar/>). La estructura de la carpeta se muestra en la Figura 4.36.

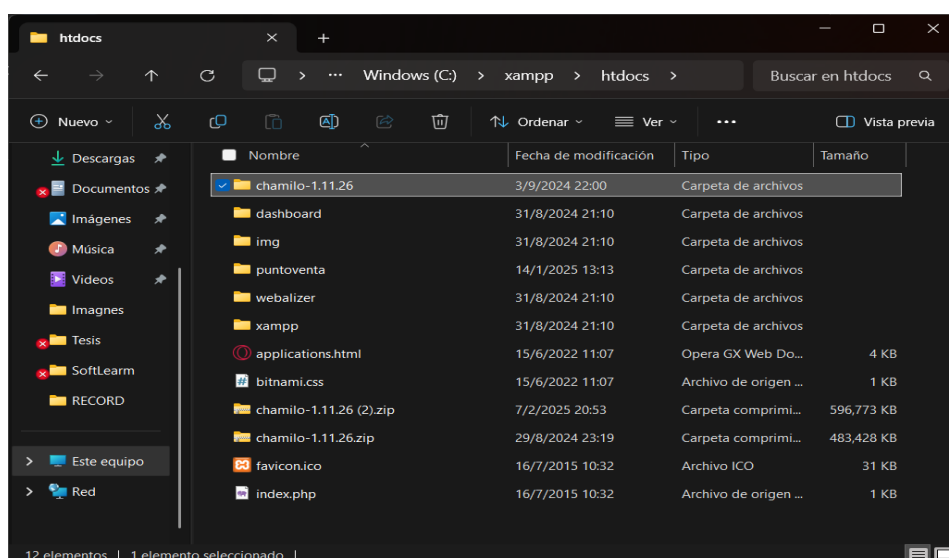


Figura 4.36. Captura de pantalla de la carpeta htdocs.

- Al ingresar a XAMPP se activan los módulos de “Apache” y “MySQL”, los cuales cambian a color verde cuando están activos de forma correcta.
- Se ingresa al botón “Admin” del módulo Apache, lo que abrirá el navegador por defecto y se muestra la pantalla principal de XAMPP y luego se cambia la dirección URL por <http://localhost/chamilo-1.11.26/> para iniciar la instalación del LMS.
- Se ingresa al apartado de trabajo de phpMyAdmin, ingresando desde el botón “Admin” del apartado de MySQL. A continuación se crea una base de datos con el nombre de “Chamilo”.
- Se realiza la instalación de Chamilo siguiendo las instrucciones y asignando la base de datos creada anteriormente, como podemos observar en la Figura 4.37.

El script de instalación creará las principales bases de datos de Chamilo. Por favor, recuerde que Chamilo necesitará crear varias bases de datos. Si sólo puede tener una base de datos en su proveedor, Chamilo no funcionará.

Servidor de base de datos ej. localhost

Puerto ej. 3306

Nombre de usuario de la base de datos ej. root

Contraseña de la base de datos ej. Kfk77rLu

Base de datos principal de Chamilo (BD)

[Comprobar la conexión con la base de datos](#)

Database host: localhost
Database port: 3306
Database driver: pdo_mysql

2. Requerido
3. Licencia
4. **Parámetros de las bases de datos MySQL**
5. Parámetros de configuración
6. Sumario de la instalación
7. Instalar

[Leer la guía de instalación](#)

Figura 4.37. Captura de pantalla de la configuración sobre la base de datos local (Chamilo, 2024).

- Al finalizar se visualiza la interfaz principal de que Chamilo se instaló correctamente.
 - Se personaliza la interfaz, se configura todo el curso de Chamilo (lecciones, actividades y recursos adicionales) y al finalizar se realizan las pruebas iniciales.
2. Adquisición de dominio y hosting en HostGator:
- Después de tener el proyecto de forma local se busca disponer el curso al alcance de todos. Por esta razón, se adquiere un dominio a través de del proveedor HostGator. A continuación se muestra el procedimiento:

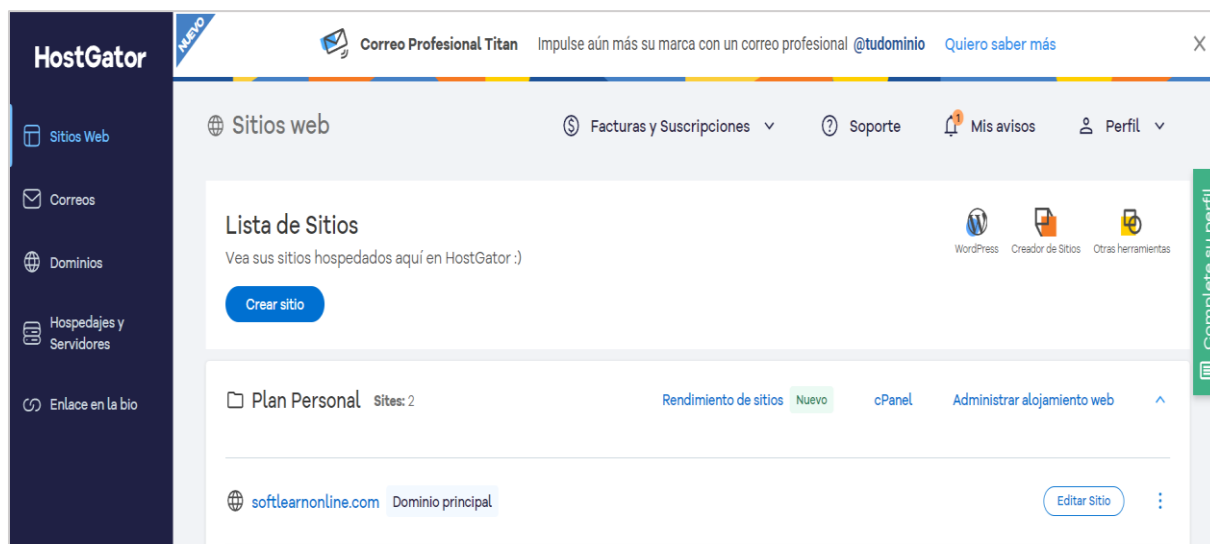


Figura 4.38. Panel de control de HostGator (HostGator).

- Contratación de servicios de hosting y dominio en HostGator, se ingresa a la cuenta de Hostinger y se accede al panel de control, como se muestra en Figura 4.38.

- Se realiza la configuración básica del servidor.

3. Transferencia de archivos al servidor mediante FileZilla:

- Migración de Chamilo con una copia de seguridad del entorno local al servidor utilizando FileZilla, como se observa en la Figura 4.39.

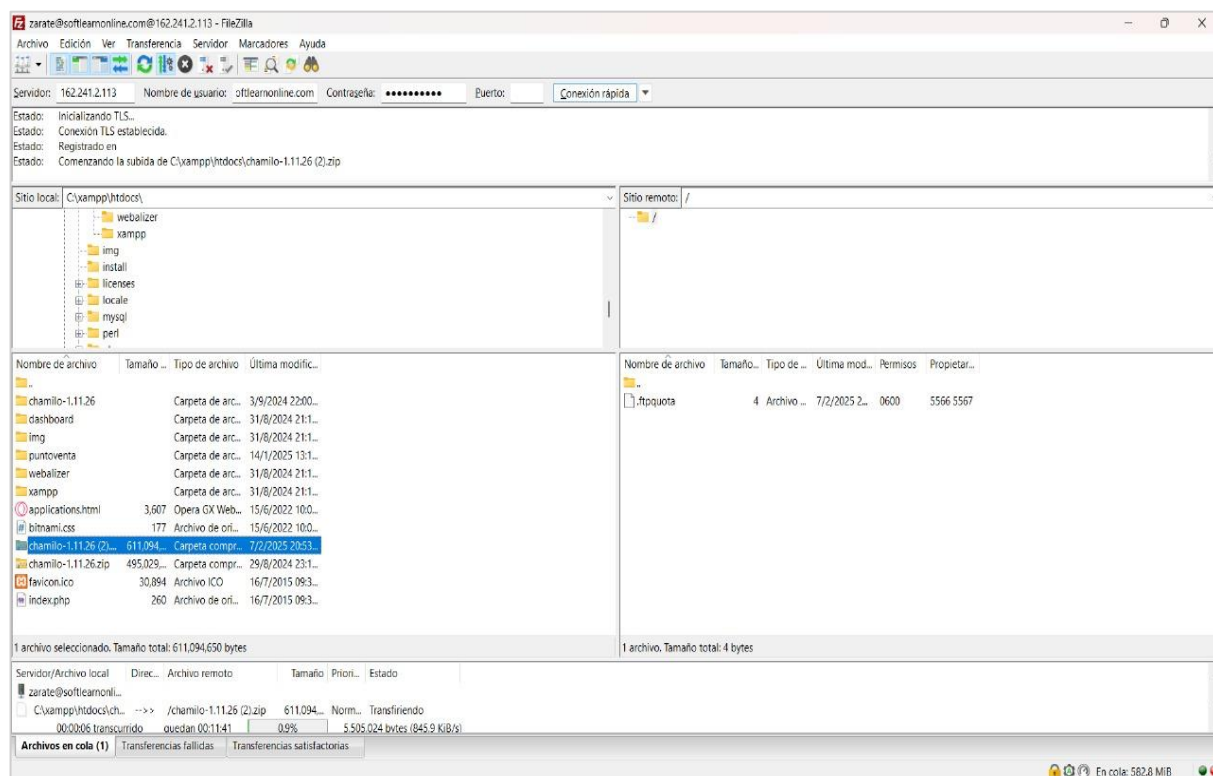


Figura 4.39. Conexión del servidor desde FileZilla.

- Se verifica la carga correcta de archivos y permisos.

4. Migración de la base de datos:

- Se crea una base de datos desde el cPanel del dominio, luego se crea un usuario con su respectiva contraseña, finalmente se asocia el usuario creado anteriormente con la base de datos.
- Se importa la base de datos MySQL al servidor desde el apartado de phpMyAdmin en “Importar”, como se muestra en la Figura 4.40.

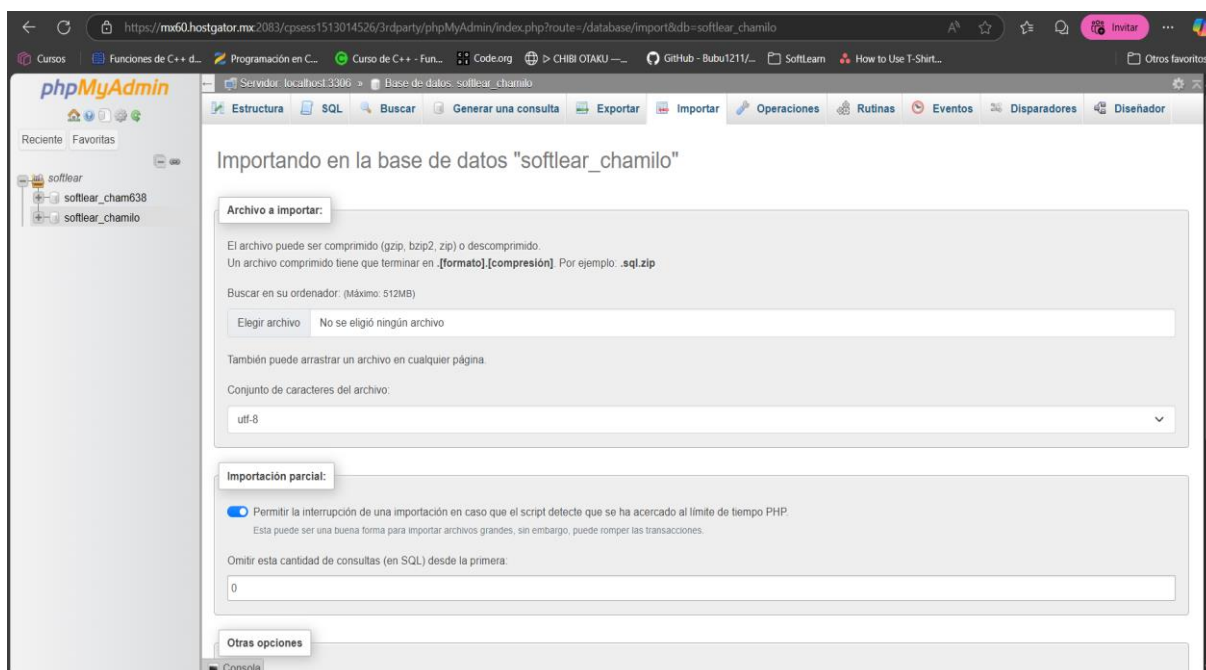


Figura 4.40. Importación de base de datos “softlear_chamilo” (HostGator).

- Ajuste del archivo de configuración de Chamilo para reconocer la nueva base de datos.
- #### 5. Pruebas de funcionalidad y de carga de contenido:
- Verificación de la ejecución de Chamilo en línea, se comprueba que la plataforma Chamilo funcione correctamente al ingresar desde su dominio <https://softlearnonline.com/> como se observa en la Figura 4.41.
 - Para poder ingresar será necesario crear un usuario, donde te piden un correo y contraseña.
 - Se realizan pruebas de módulos de aprendizaje y actividades, para garantizar que todo se visualice y funcione correctamente.

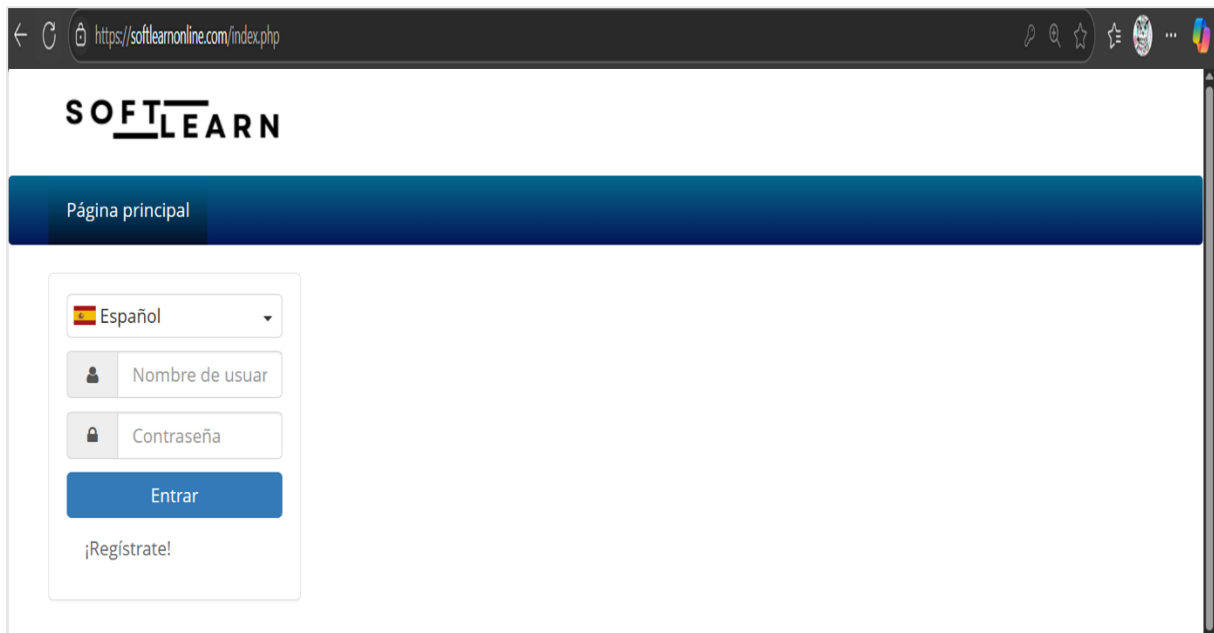


Figura 4.41. Interfaz del dominio SoftLearn.

4.5 Evaluación

La evaluación de SoftLearn es una fase crucial para asegurar que el SPOC cumpla con estándares adecuados de usabilidad, accesibilidad y experiencia del usuario. Por ello, se usa la herramienta SUS (*System Usability Scale*), desarrollada por John Brooke en 1986, que permite medir la usabilidad de plataformas, cursos en línea, sitios web y sistemas en general.

Esta escala fue diseñada como un instrumento breve y confiable para medir la percepción de usabilidad de un sistema por parte de los usuarios. Se basa en la definición de usabilidad establecida por la norma ISO 9241-11, que considera tres puntos: eficacia (logro de objetivos), eficiencia (uso óptimo de recursos) y satisfacción (grado de comodidad o agrado durante el uso) (Hedlefs Aguilar & Garza Villegas , 2016).

La encuesta SUS se aplica a una muestra representativa del público objetivo del curso, dicha evaluación permitirá identificar fortalezas y debilidades relacionadas con la navegación, accesibilidad, claridad de la interfaz y experiencia general de uso. Los resultados permitirán establecer acciones de mejora orientadas tanto al diseño pedagógico como a la optimización técnica del entorno.

La encuesta se diseñó y aplicó a través de *Google Forms*, lo que facilita la recolección y el análisis de datos en un entorno accesible. A continuación se presentan la Figura 4.42 que muestra el encabezado de la encuesta y la Figura 4.43 una muestra de las preguntas realizadas:



Encuesta de Evaluación del Curso SoftLearn

Gracias por participar en la evaluación de SoftLearn, el curso de Ingeniería de Software enfocado en Análisis de Sistemas de Software. Tu opinión es muy importante para mejorar la calidad del contenido, la metodología y la experiencia en la plataforma. Por favor, responde las siguientes preguntas utilizando la escala propuesta.

Figura 4.42. Encabezado de la encuesta.

Evaluación de Usabilidad del Curso
Instrucciones: Marca una opción según tu nivel de acuerdo con cada afirmación.

1 = Totalmente en desacuerdo
 2 = En desacuerdo
 3 = Ni de acuerdo ni en desacuerdo
 4 = De acuerdo
 5 = Totalmente de acuerdo

1. Me gustaría usar esta plataforma con frecuencia para otros cursos. *

1 2 3 4 5

Totalmente en desacuerdo ☐ ☐ ☐ ☐ ☐ Totalmente de acuerdo

2. Considero que el sistema presenta demasiada complejidad en su uso. *

1 2 3 4 5

Totalmente en desacuerdo ☐ ☐ ☐ ☐ ☐ Totalmente de acuerdo

3. Encontré la plataforma SoftLearn fácil de usar. *

1 2 3 4 5

Totalmente en desacuerdo ☐ ☐ ☐ ☐ ☐ Totalmente de acuerdo

4. Creo que necesitaría el apoyo de un experto para poder utilizar esta plataforma. *

1 2 3 4 5

Totalmente en desacuerdo ☐ ☐ ☐ ☐ ☐ Totalmente de acuerdo

5. Las funcionalidades del curso están bien integradas y permiten una navegación fluida. *

1 2 3 4 5

Totalmente en desacuerdo ☐ ☐ ☐ ☐ ☐ Totalmente de acuerdo

Figura 4.43. Muestra de preguntas primera sesión.

3.5.1 Análisis de resultados SUS

La evaluación de la usabilidad de SoftLearn se realizó utilizando el cuestionario SUS, compuesto por 10 preguntas estándar. Las respuestas se recogen mediante una escala de 1 a 5, donde 1 corresponde a “totalmente en desacuerdo”, 2 a “en desacuerdo”, 3 a “ni de acuerdo ni en desacuerdo”, 4 a “de acuerdo” y 5 a “totalmente de acuerdo”.

Una vez que los estudiantes han completado la encuesta, se procede al análisis de los datos recopilados. Esta herramienta ofrece datos cuantificables que proporciona un índice claro de la usabilidad del sistema, facilitando la identificación de fortalezas y áreas de mejora en aspectos clave como la plataforma, la navegación, accesibilidad y la experiencia de uso en general.

A continuación, se presentan los resultados individuales de cada pregunta del cuestionario SUS, acompañados de gráficos que muestra la distribución de respuestas en cada una de las preguntas, lo que permite una mejor visualización de los resultados:

En la Figura 4.44, se muestran los resultados de la pregunta 1, donde el 90% de los encuestados indicaron estar “Totalmente de acuerdo” en que les gustaría utilizar la plataforma con frecuencia para otros cursos.

Un 10% de los participantes seleccionó la opción “De acuerdo”, mientras que ninguna persona eligió las opciones “Totalmente en desacuerdo”, “En desacuerdo” y “Ni de acuerdo ni en desacuerdo”.

Estos resultados muestran un alto nivel de aceptación y satisfacción con la plataforma.

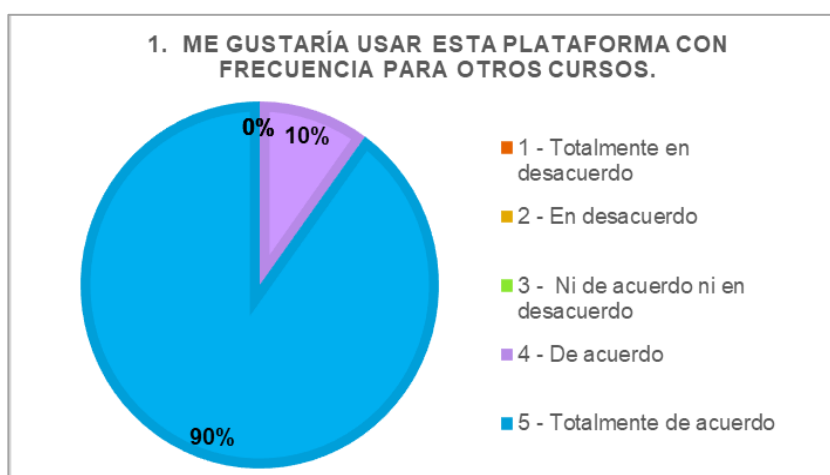


Figura 4.44. Gráfico de resultados pregunta 1

La Figura 4.45 muestra los resultados de la pregunta 2, donde el 45% de los encuestados seleccionó “Totalmente en desacuerdo” y el 35% “En desacuerdo”, lo que indica que la mayoría de los participantes considera que el sistema no presenta complejidad en su uso. Un 10% indicó “De acuerdo”, mientras que otro 10% seleccionó “Totalmente de acuerdo”. Además, ningún participante optó por la opción “Ni de acuerdo ni en desacuerdo”.

Estos resultados muestran que la percepción general de que el sistema es eficiente de usar y no presenta dificultades significativas en su manejo.

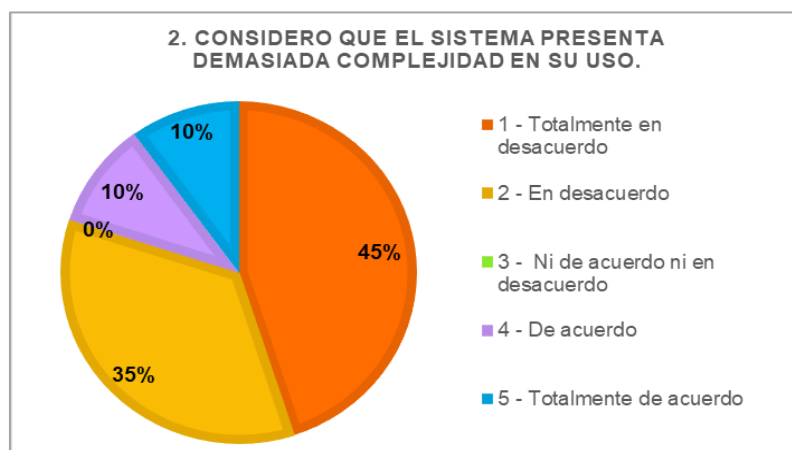


Figura 4.45. Gráfico de resultados pregunta 2

Las respuestas correspondientes a la pregunta 3 se observan en la Figura 4.46, donde un 75% de los encuestados indicó estar “Totalmente de acuerdo” con que SoftLearn es fácil de usar, mientras que un 20% selecciono la opción “De acuerdo”. Solo un 5% manifestó estar “Totalmente en desacuerdo”, mientras que ninguna persona eligió las opciones “En desacuerdo” o “Ni de acuerdo ni en desacuerdo”.

Estos resultados reflejan una percepción mayoritariamente positiva respecto a la facilidad de uso de la plataforma.

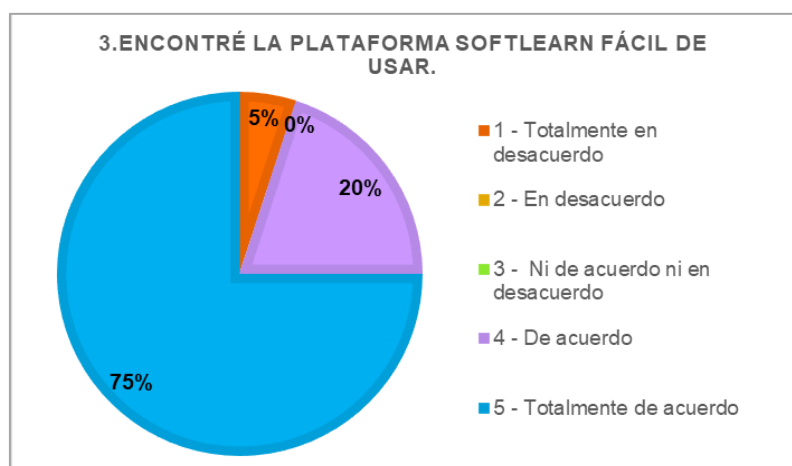


Figura 4.46. Gráfico de resultados pregunta 3

La Figura 4.47 muestra las respuestas obtenidas para la pregunta 4: “Creo que necesitaría el apoyo de un experto para utilizar esta plataforma”. Un 55% de los encuestados indicó estar “Totalmente en desacuerdo”, mientras que un 25% seleccionó la opción en “En desacuerdo”. Además, un 10% de los encuestados se mostró neutral, mientras que un 5% indicó estar “De acuerdo” y otro 5% seleccionó “Totalmente de acuerdo”.

Estos resultados reflejan una percepción positiva respecto a la facilidad de uso de la plataforma, ya que la mayoría de los encuestados no considera necesario un apoyo.

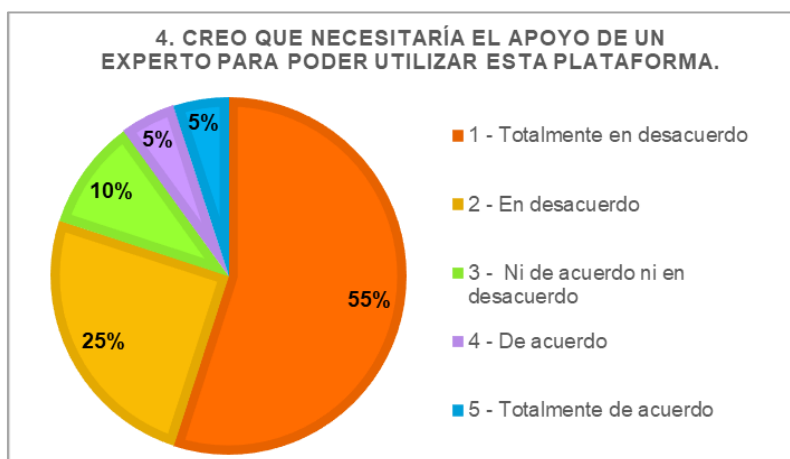


Figura 4.47. Gráfico de resultados pregunta 4

Los resultados de la pregunta 5 se observan en la Figura 4.48, donde un 85% de los encuestados indicó estar “Totalmente de acuerdo” en que las funcionalidades del curso están bien integradas y permiten una navegación fluida, mientras que un 15% selecciono la opción “De acuerdo”. Ningún encuestado manifestó estar en desacuerdo o una opinión neutral.

Lo que refleja una percepción mayormente positiva sobre la integración y facilidad de navegación del curso.

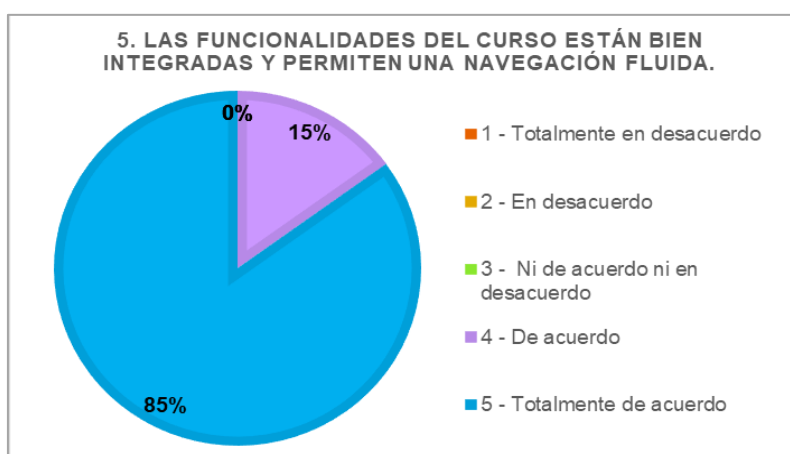


Figura 4.48. Gráfico de resultados pregunta 5

Para la pregunta 6, los resultados se detallan en la Figura 4.49, donde un 85% de los encuestados indicó estar “Totalmente en desacuerdo” con que experimentaron inconsistencias o problemas técnicos, mientras que un 10% selecciono la opción “En desacuerdo”. El 5% restante se mostró neutral, seleccionando “Ni de acuerdo ni en desacuerdo”, mientras que ninguna persona eligió las opciones “De acuerdo” o “Totalmente de acuerdo”.

Estos resultados reflejan una percepción mayormente positiva respecto a la estabilidad técnica de la plataforma, ya que la gran mayoría no reporto problemas.

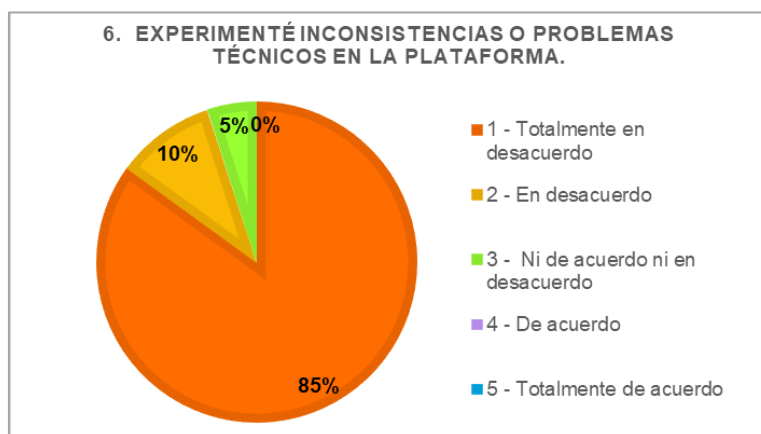


Figura 4.49. Gráfico de resultados pregunta 6

En la Figura 4.50 se observan las respuestas de la pregunta 7: “¿Creo que la mayoría de los estudiantes aprenderían a usar esta plataforma rápidamente?”. Un 95% de los encuestados indicó estar “Totalmente de acuerdo”, mientras que un 5% seleccionó la opción “De acuerdo”. Además, no hubo respuestas en desacuerdo ni opiniones neutrales.

Lo que refleja una percepción altamente positiva sobre la facilidad de aprendizaje y uso del curso.

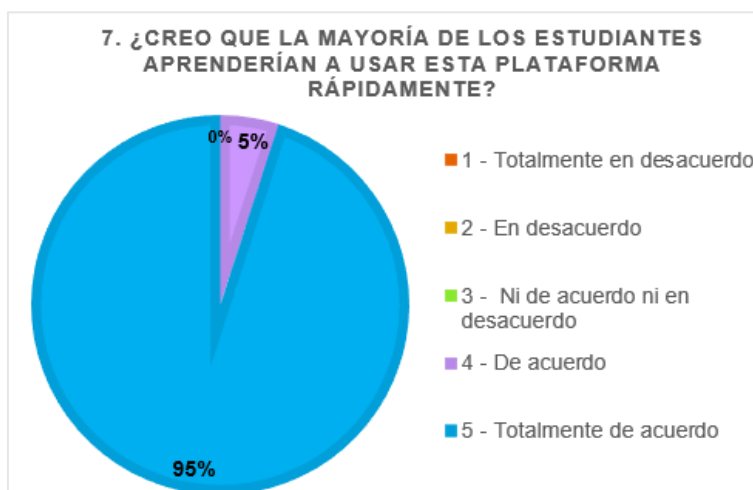


Figura 4.50. Gráfico de resultados pregunta 7

La representación gráfica de la pregunta 8 se muestra en la Figura 4.51, donde un 70% de los encuestados indicó estar “Totalmente de acuerdo” en que las funciones y herramientas del curso están bien organizadas y son accesibles, mientras que un 25% seleccionó “De acuerdo”. El 5% restante seleccionó “Totalmente en desacuerdo”, mientras que ninguna persona eligió las opciones “En desacuerdo” o “Ni de acuerdo ni en desacuerdo”.

Estos resultados reflejan una percepción mayormente positiva sobre la organización y accesibilidad del curso.

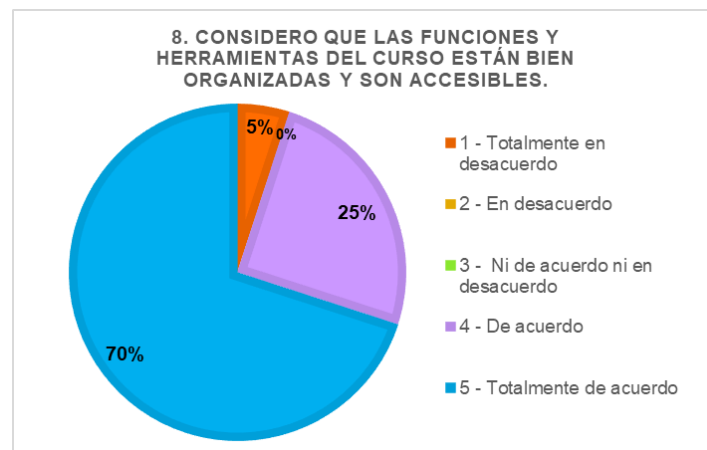


Figura 4.51. Gráfico de resultados pregunta 8

Los datos obtenidos para la pregunta 9 pueden observarse en la Figura 4.52, donde un 90% de los encuestados indicó estar “Totalmente de acuerdo” en que se sintieron seguros y confiados al navegar por el curso y acceder a sus recursos, mientras que el otro 10% seleccionó “De acuerdo”. No se registraron respuestas en las demás categorías: “Totalmente en desacuerdo”, “En desacuerdo” o “Ni de acuerdo ni en desacuerdo”.

Esto indica una percepción mayormente positiva respecto a la confianza y seguridad al usar la plataforma.

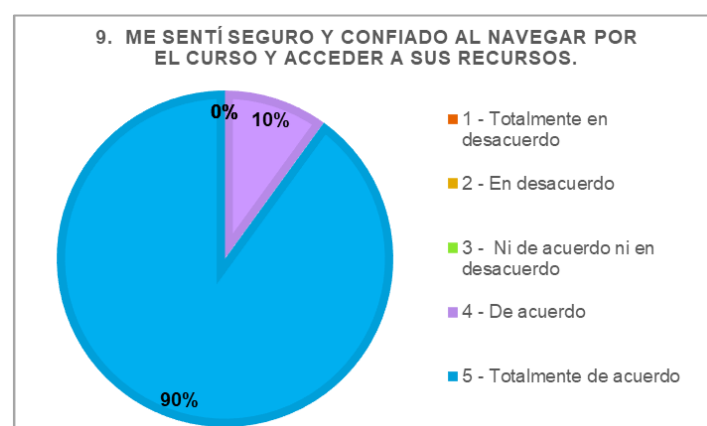


Figura 4.52. Gráfico de resultados pregunta 9

La Figura 4.53 muestra los resultados de la Pregunta 10, donde un 80% de los encuestados indicó estar “Totalmente de acuerdo” en que la interfaz de usuario del curso es intuitiva y agradable, mientras que un 20% seleccionó la opción “De acuerdo”. No se registraron respuestas en las demás categorías: “Totalmente en desacuerdo”, “En desacuerdo” o “Ni de acuerdo ni en desacuerdo”.

Lo que refleja una opción mayormente positiva respecto a la usabilidad y atractivo de la interfaz del curso.

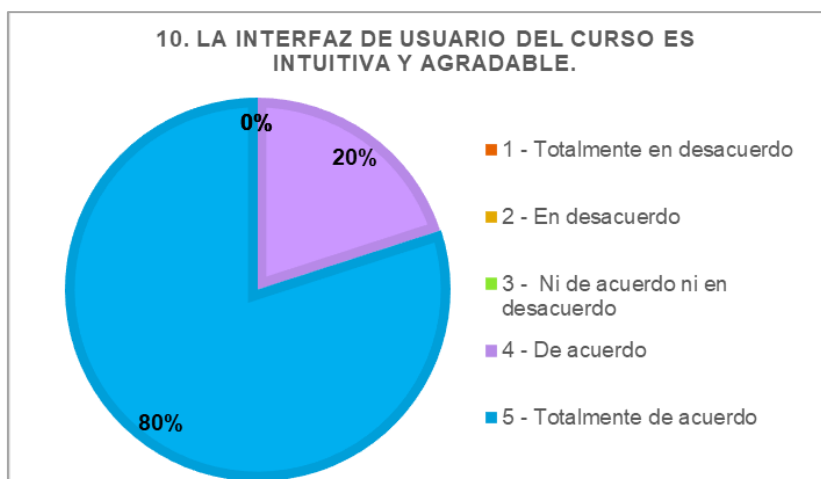


Figura 4.53. Gráfico de resultados pregunta 10

3.5.2 Cálculo del puntaje SUS

El cálculo del puntaje de usabilidad para el curso se lleva a cabo mediante los siguientes pasos (Cordero de la Cruz, 2020):

1. Para las preguntas impares (1, 3, 5, 7, 8, 9, 10):
Se resta 1 a la puntuación proporcionada por el usuario (por ejemplo, si el usuario responde con un 5, se resta 1, obteniendo una puntuación ajustada de 4).

Nota: Aunque las preguntas 8 y 10 son pares, en la encuesta realizada se formuló de manera positiva estas preguntas, lo que hace necesario que se ajusten dentro de este grupo para mantener la coherencia en el proceso de evaluación.

2. Para las preguntas pares (2, 4, 6):
Se resta la puntuación de 5 a la puntuación brindada por el usuario (por ejemplo, a el valor 5 se le resta 4 que fue el número que seleccionó el usuario, obteniendo una puntuación final de 1).

3. Se realiza una suma de las puntuaciones ajustadas para cada encuesta completada.
4. Se multiplica el total por 2.5 para obtener un puntaje final que varía de 0 a 100, que indica la usabilidad general del sistema.

En la Figura 4.54, se muestra cómo interpretar los resultados obtenidos con el cálculo del puntaje SUS (Arias Del Prado, 2024):

- Puntuación baja (0-50): Es necesario realizar cambios importantes en la usabilidad del sistema.
- Puntuación media (51-70): La usabilidad del sistema es marginal, pero se pueden realizar mejoras.
- Puntuación alta (71-100): La usabilidad del sistema es aceptable, pero siempre hay espacio para la mejora continua.

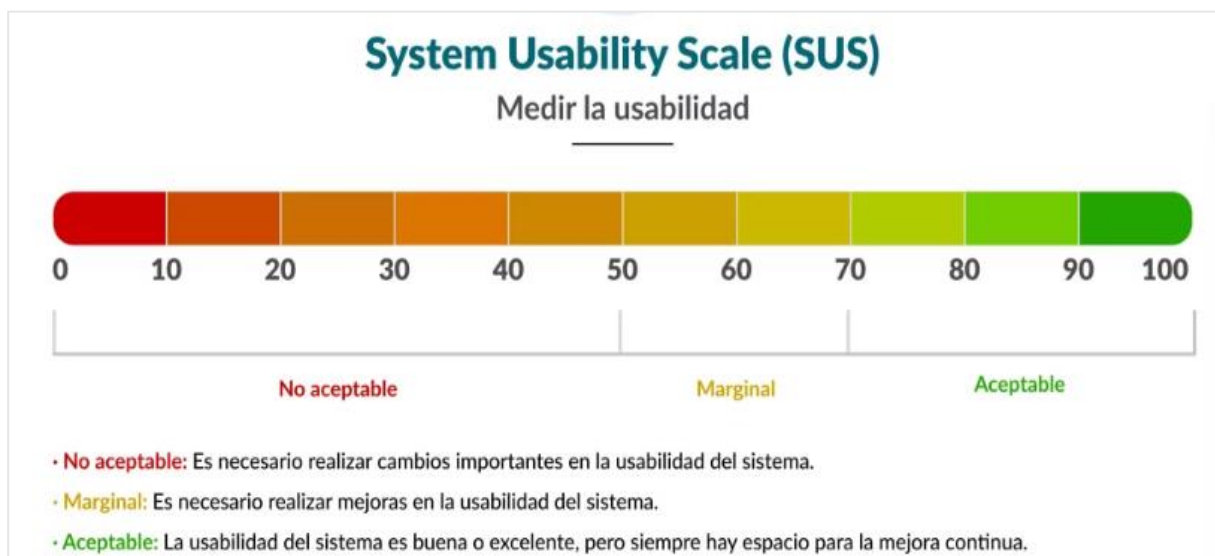


Figura 4.54. Interpretación de resultados SUS (Arias Del Prado, 2024)

Evaluación de los datos de las encuestas realizadas:

De acuerdo con los datos obtenidos en la encuesta y después de realizar el cálculo del puntaje SUS, los resultados obtenidos son los siguientes:

- Puntaje mínimo SUS: 70
- Puntaje máximo SUS: 100
- Promedio SUS: 92.5

Según los datos demostrados en la Figura 4.55, se deduce lo siguiente:

- El 90% de los encuestados calificaron la usabilidad como excelente, con puntajes entre 80 y 100 en SUS, es decir que la plataforma es altamente funcional y fácil de usar.
- Solo el 10% calificó la experiencia con una “usabilidad buena”, con un puntaje de 70 o 80.
- No hubo encuestas con resultados por debajo de un puntaje de 70, lo que indica una usabilidad del curso muy positiva.

# encuesta	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Suma total	SUS
1	4	3	4	3	4	3	4	3	4	3	35	87.5
2	4	3	0	4	4	4	4	4	4	4	35	87.5
3	4	0	4	3	4	4	4	4	4	4	35	87.5
4	4	4	4	4	4	4	4	4	4	4	40	100
5	4	4	3	3	4	4	4	4	4	4	38	95
6	3	4	4	4	4	4	4	4	4	4	39	97.5
7	4	1	4	2	4	4	4	4	4	4	35	87.5
8	4	4	4	3	4	4	4	3	4	3	37	92.5
9	4	3	3	1	4	4	4	3	4	3	33	82.5
10	4	3	3	4	4	2	3	4	4	4	35	87.5
11	3	4	4	4	3	4	4	3	4	4	37	92.5
12	4	4	4	4	4	4	4	4	4	4	40	100
13	4	3	4	4	4	4	4	4	4	4	39	97.5
14	4	3	4	4	4	4	4	4	4	4	39	97.5
15	4	4	4	4	4	4	4	4	4	4	40	100
16	4	1	4	0	3	3	4	3	3	3	28	70
17	4	0	4	2	3	4	4	4	3	4	32	80
18	4	4	4	4	4	4	4	0	4	4	36	90
19	4	4	4	4	4	4	4	4	4	4	40	100
20	4	3	3	3	4	4	4	4	4	4	37	92.5

Figura 4.55. Evaluación de resultados SUS. Fuente: Resultados de encuesta

Conclusiones

En las últimas décadas, el desarrollo educativo ha estado marcado por el impacto de las tecnologías digitales. Herramientas en línea, entornos virtuales de aprendizaje y recursos interactivos han permitido una transformación en la forma en que los estudiantes adquieren conocimientos, favoreciendo una enseñanza más dinámica y adaptativa. Esta transformación ha sido fundamental para evolucionar el aprendizaje tradicional, al incorporar modalidades más flexibles, personalizadas y accesibles que contribuyen al desarrollo de competencias clave.

En este contexto, SoftLearn ha demostrado ser una solución efectiva para la enseñanza del Análisis de Sistemas de Software en el área de Ingeniería de Software. Este curso no solo brinda un enfoque estructurado y modular para enseñar conceptos fundamentales, sino que también proporciona una plataforma interactiva en línea que facilita el aprendizaje autónomo, al permitir que cada estudiante avance a su propio ritmo, desde cualquier lugar y cualquier momento.

El diseño pedagógico de SoftLearn se estructura en base a la metodología ADDIE, una estructura metodológica reconocido por su enfoque sistemático para diseñar experiencias de aprendizaje efectiva. Esta metodología permitió organizar el curso en módulos progresivos, con una presentación clara y secuencial de los conceptos clave del Análisis de Sistemas de Software. Además, facilitó la incorporación de retroalimentación constante, fortaleciendo la comprensión de los conceptos y su aplicación.

La metodología ADDIE también permitió la integración favorable de herramientas tecnológicas clave para el desarrollo de SoftLearn. La plataforma Chamilo utilizada como LMS, que proporciona la estructura necesaria para la gestión del contenido, la administración del curso y creación de ejercicios, al igual que HotPotatoes, que facilitó la creación de ejercicios interactivos orientados a la evaluación continua. Estas tecnologías no solo mejoran la experiencia educativa, al estar en un sitio web, contribuyen a una experiencia educativa dinámica, accesible desde cualquier dispositivo con conexión a internet, lo que amplía las posibilidades de estudio y refuerzo fuera del aula.

La evaluación del curso, realizado mediante la escala SUS, arrojó un puntaje promedio de 92.5%, lo que refleja un alto nivel de satisfacción por parte de los usuarios. Los participantes destacaron la facilidad de navegación, claridad en la organización del contenido y la inductividad de la interfaz. Estos resultados confirman que SoftLearn no solo responde a una necesidad real en la formación académica de estudiantes de Ingeniería de Software, sino que también cumple con las expectativas tanto en términos de usabilidad como de funcionalidades.

En resumen, SoftLearn ha cumplido con los objetivos establecidos, al ofrecer una plataforma educativa accesible e interactiva que fomenta el aprendizaje autónomo de los estudiantes, gracias a la aplicación del modelo ADDIE, se estructuro el curso que integra de manera efectiva los aspectos pedagógicos y tecnológicos, creando un entorno de aprendizaje eficiente y bien diseñado. Además, demuestra el potencial de las plataformas en línea para transformar la educación, al facilitar experiencias de aprendizaje más accesibles e interactivas.

Trabajos futuros

SoftLearn ha sido desarrollado con un enfoque inicial centrado en el Análisis de Sistemas de Software para estudiantes de Ingeniería de Software. Sin embargo, existen diversas áreas que podrían expandirse y mejorar, ofreciendo oportunidades para enriquecer el curso. Con base en este enfoque, se identifican varias propuestas para continuar con el desarrollo y la mejora del proyecto.

Una posible ampliación del contenido del curso para mejorar su impacto educativo, el curso podría expandirse para incluir más módulos sobre las etapas de Ingeniería de Software, tales como diseño de sistemas software, desarrollo de sistemas de software, gestión de proyectos software, pruebas de software y metodologías ágiles. Esta ampliación no solo fortalecer el contenido, sino que también permitirá una formación más completa al abarcar más aspectos de la disciplina.

Por otro lado, sería beneficioso la incorporación de nuevas herramientas educativas que se integren con Chamilo para la creación de ejercicios interactivos más diversos. Esto permitirá variar los tipos de ejercicios y mejorar la experiencia de aprendizaje.

Además, gracias a la flexibilidad y escalabilidad de Chamilo como LMS, SoftLearn tiene el potencial de incluir otros cursos ya sean relacionados con la Ingeniería de Software u otras relacionadas con la Licenciatura en Ciencias Computacionales. Esta expansión podría apoyar a los estudiantes durante su recorrido en la carrera.

Referencias

- Aguayo Sarasa, R., & Bravo-Agapito, J. (2017). Implantación de un SPOC en la educación a distancia para la mejora del proceso de enseñanza-aprendizaje. *Revista Tecnología Ciencia y Educación*, 6, 129-142. <https://doi.org/10.51302/tce.2017.119>
- Aguilera, C. (7 de julio de 2024). ispring. Guía completa del modelo ADDIE para el e-learning. <https://www.ispring.es/blog/modelo-addie>
- Allen, M. (2012). Leaving ADDIE for SAM: An agile model for developing the best learning experiences. American Society for Training & Development. American Society for Training & Development. <https://archive.org/details/leavingaddiefors0000alle/page/n1/mode/1up>
- Arias Del Prado, J. (17 de abril de 2024). Medir con el sistema de escala de usabilidad (SUS). UXABLES. <https://www.uxables.com/investigacion-ux/medir-con-el-sistema-de-escala-de-usabilidad-sus/>
- Arizala Campo, R. E., Chalco López, C. L., Birmania Piedad, M. L., & Paucar Ñacata, V. P. (2023). Impacto De Las Plataformas Digitales En El Aprendizaje Colaborativo: Análisis De Casos Y Prácticas Exitosas. *Ciencia Latina Revista Científica Multidisciplinar*, 7(3), 1321-1342. https://doi.org/10.37811/cl_rcm.v7i3.6275
- Blackboard Inc. (2025). Mi Blackboard. Blackboard Help. <https://help.blackboard.com/es-es/node/38241>
- Canva. (2021). Acerca de Canva. Canva. https://www.canva.com/es_es/about/
- Cañellas Mayor, A. (2011). CMS, LMS y LCMS. Definición y diferencias. *Revista Comunicación y Pedagogía: nuevas tecnologías y recursos didácticos*, 251-252(1136-7733), 16-18.
- Castro, S., Clarenc, C., López de Lenz, C., Moreno, M., & Tosco, N. B. (2013). Analizamos 19 plataformas de e learning primera investigacion academica colaborativa mundial. Lulu. Analizamos 19 plataformas de e learning primera investigacion academica colaborativa mundial. <https://www.calameo.com/read/004216390e72a30dae401>
- Chamilo. (2020). ¿Qué es Chamilo? chamilo.org. https://docs.chamilo.org/v/1.11.x-es/manual-del-profesor/introduccion/que_es_chamilo
- Chamilo. (2024). Descargar Chamilo. Chamilo.org. <https://chamilo.org/es/descargar/>
- Coaguila Mayanaza, D. V., García Palomino, R., & Cruz Arpi, F. N. (24 de marzo de 2023). Oportunidades y desafíos de la educación híbrida en el contexto pospandémico. *Horizontes. Revista de Investigación en Ciencias de la Educación*, 7(28, Ed. Extraordinaria), 1028-1041. <https://doi.org/https://doi.org/10.33996/revistahorizontes.v7i28.572>

- Colman, H. (20 de noviembre de 2024). SCORM: Qué es y cómo usarlo. Blog de E-learning. ispring. <https://www.ispring.es/blog/que-es-scorm>
- Cordero de la Cruz, J. E. (26 de marzo de 2020). Cuestionario para Medir El Uso de La Usabilidad. UXtweak. <https://es.scribd.com/document/506386138/Cuestionario-Para-Medir-El-Uso-de-La-Usabilidad>
- Coursera. (2012). CourseraRa. Degrees, Certificates, & free online course. <https://www.coursera.org>
- Defense Technical Information Center. (1975). Internet Archive. Interservice Procedures for Instructional Systems Development. Executive Summary and Model. https://archive.org/details/DTIC_ADA019486/page/n4/mode/1up
- Díaz Quilla, J. P., Carbonel Alta, G. Z., & Picho Durand, D. J. (2021). Los Sistemas de Gestión de Aprendizaje (LMS) en la educación virtual. Revista arbitrada del centro de investigación y estudios gerenciales, 50(Julio - Agosto 2021), 87-95. <https://revista.grupociieg.org/wp-content/uploads/2021/06/Ed.5087-95-Diaz-Carbonel-Picho.pdf>
- edX. (2023). edX. edX. <https://home.edx.org>
- edX. (2024). UAMx: Introducción a la Ingeniería del Software. edX. <https://www.edx.org/learn/software-engineering/universidad-autonoma-de-madrid-introduccion-a-la-ingenieria-del-software>
- Flores Tena, M. J., Ortega Navas, M. d., & Sánchez Fuster, M. C. (2021). Las nuevas tecnologías como estrategias innovadoras de enseñanza aprendizaje en la era digital. Revista Electrónica Interuniversitaria de Formación del Profesorado, 24(1), 29-42. <https://doi.org/10.6018/reifop.406051>
- Gagné, R. (1965). The conditions of learning. Holt, Rinehart and Winston en Nueva York. https://archive.org/details/conditionsoflear0000gagn_d4x8/page/n7/mode/1up
- Google. (s.f.). Google Digital Garage. Google. https://skillshop.exceedlms.com/student/catalog/list?category_ids=7879&locale=en-GB
- Hedlefs Aguilar, M. I., & Garza Villegas, A. A. (2016). Análisis comparativo de la Escala de Usabilidad del Sistema (EUS) en dos versiones. Revista Iberoamericana de las Ciencias Computacionales e Informática, 5(10), 1-17.
- Heinich, R., Molenda, M., Russell, J., Smaldino, S., & Hackbarth, S. (1999). Instructional Media and Technologies for Learning. Pearson College Div. Instructional Media and Technologies for Learning. <https://ingridrodriguez.wordpress.com/wp-content/uploads/2010/08/modelo-assure.pdf>
- Herrera, A. (2024). innovscionycualificacion. MOOC, CMOOC, XMOOC, SPOC y NOOC [principales diferencias].

<https://www.innovacionycualificacion.com/contenidos-elearning/diferencias-cursos-mooc-xmooc-cmooc-y-spoc/>

HostGator. (s.f.). Portal del Cliente. HostGator México. HostGator México. <https://cliente.hostgator.mx/inicio>

HotPotatoes. (2005). Hot Potatoes Home Page. hotpot. <https://hotpot.uvic.ca>

Ibarra-Corona, M. A., & Escudero-Nahón, A. (2021). Metasíntesis sobre la aplicación de principios de Ingeniería de Software en el desarrollo de plataformas de tecnología educativa. *Revista Interuniversitaria de Investigación en Tecnología Educativa (RIITE)*(10), 62-75. <https://doi.org/10.6018/riite.463421>

Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF). (2025). ¿Qué es un NOOC? EducaLAB. <http://educalab.es/intef/formacion/formacion-en-red/nooc#:~:text=Un%20NANO%20Curso%20Abierto%2C%20Masivo%20y%20En%20L%C3%ADnea,horas%20de%20esfuerzo%20estimado%20de%20dificultad%20al%20NOOC.>

Instituto Tecnológico y de Estudios Superiores de Monterrey. (2023). Canvas | Tecnologías para la Educación. Tecnológico de Monterrey. <https://tedu.tec.mx/es/canvas>

ISO/IEC. (2018). ISO/IEC 19796-1:2005 - Information technology — Learning, education and training — Quality management, assurance and metrics — Part 1: General approach. ISO. <https://www.iso.org/standard/33934.html>

ISO/IEC. (2023). ISO/IEC 25010:2023 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model (2ª ed.). ISO. <https://www.iso.org/cms/%20render/live/es/sites/isoorg/contents/data/standard/07/81/78176.html>

ISO/IEC/IEEE. (2020). ISO. ISO/IEC/IEEE 12207-2:2020 - Systems and software engineering — Software life cycle processes — Part 2: Relation and mapping between ISO/IEC/IEEE 12207:2017 and ISO/IEC 12207:2008. <https://www.iso.org/standard/73446.html>

Laguna Lozano, M. (2024). Introducción al Modelo de Referencias SCORM. Área de Tecnologías para la Docencia Universidad Zaragoza. https://www.unpa.edu.ar/sites/default/files/descargas/Coberturas%20NODOCENTES/4.%20Materiales/2015/T093/SCORM_Standar.pdf

LinkedIn. (s.f.). LinkedIn Learning. LinkedIn. <https://www.linkedin.com/learning/?originalSubdomain=es>

López de la Serna, A., Castaño Garrido, C., & Herrero Fernández, D. (2018). Integración de los cursos SPOC en las asignaturas de grado. *Pixel-Bit: Revista de medios y educación*(52), 139-149.

- López Ortega, O., Ramos Fernández, J., Ortega Mendoza, R., & Márquez Vera, M. (agosto de 2022). Editorial del número especial en conmemoración del XX aniversario del doctorado en ciencias computacionales. semanticscholar. <https://www.semanticscholar.org/paper/Editorial-del-número-especial-en-conmemoración-del-López-Ortega-Ramos-Fernández/d14b4cdee54a2072e6f44e29d5be836e94351093>
- López, D. G. (2005). Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning. Universidad de Salamanca. https://gredos.usal.es/bitstream/handle/10366/56649/DIA_Repositoriosobjetos.pdf;jsessionid=11B748E590ECDFDC067BB5A8DB76D24A?sequence=1
- Lorenzo Garcia, A. (enero de 2020). LMS. Plataformas Virtuales o Entornos Virtuales de Aprendizaje. Ventajas y funcionalidades. ResearchGate. https://www.researchgate.net/publication/348370547_LMS_Plataformas_Virtuales_o_Entornos_Virtuales_de_Aprendizaje_Ventajas_y_funcionalidades
- Mayorga-Ases, M. J., Tagua-Moyolema, A. E., Muyulema-Muyulema, D. G., & Velastegui-Hernández, R. S. (2024). Estudio sobre la implementación de metodologías activas en la educación superior: beneficios y desafíos. 593 Digital Publisher CEIT, 196-208(2), 8. <https://doi.org/10.33386/593dp.2024.4-1.2739>
- Molenda, M. (2003). ResearchGate. In search of the elusive addie model. https://www.researchgate.net/profile/Michael-Molenda/publication/251405713_In_search_of_the_elusive_ADDIE_model/links/5b5f098ba6fdccf0b200e5b2/In-search-of-the-elusive-ADDIE-model.pdf
- Moodle. (2025). Descubre las funciones del LMS que te ayudarán a enseñar a tu manera. Moodle. <https://moodle.com/es/solutions/lms/funcionalidades/>
- Muñoz Mata, M. A., & Gasca-Hurtado, G. P. (2023). Gamificación para atender los desafíos de la enseñanza Ingeniería de Software en instituciones de educación superior. Revista Ibérica de Sistemas y Tecnologías de Información, 49(5), 5-21. <https://doi.org/10.17013/risti.49.5-21>
- NobleProg. (2025). Curso de Ingeniería de Software. NobleProg. https://www.nobleprog.com/es/cc/softeng?venue=us_24040&start-date=2025-08-04
- NobleProg. (2025). NobleProg México. NobleProg Servicios Corporativos. <https://www.nobleprog.mx>
- Organización Internacional de Normalización (ISO). (2025). ISO - Organización Internacional de Normalización. ISO. <https://www.iso.org/es/home>
- Ortega, M. C., Flores Tena, M. J., & Sánchez Fuster, M. C. (21 de enero de 2021). Revista electrónica interuniversitaria de formación del profesorado. . Las nuevas tecnologías como estrategias innovadoras de enseñanza-aprendizaje en la era digital. <https://doi.org/10.6018/reifop.406051>

- Ortiz González, L. E. (2016). ¿Qué son los MOOC? https://apps.utel.edu.mx/recursos/files/r161r/w25530w/Que_es_un_MOOC.pdf
- Pinilla Acevedo, J. C. (2016). Síntesis: MOOCs y SPOCs. Universidad Cuauhtémoc, Maestría en Educación y Entornos Virtuales de Aprendizaje. Academia. https://www.academia.edu/27607946/SINTESIS_MOOCs_Y_SPOCs
- Pozo, J. I., Cabellos, B., & Pérez Echeverría, M. d. (12 de diciembre de 2024). Has the educational use of digital technologies changed after the pandemic? A longitudinal study. PLOS ONE. <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0311695>
- Pressman, R. S. (2010). Ingeniería del software un enfoque práctico. McGraw-Hill.
- Pressman, R. S. (2010). Ingeniería del software: Un enfoque práctico. McGraw-Hill Interamericana. <https://doi.org/978-607-15-0314-5>
- Quizlet. (s.f.). Software Engineering Study Guide Flashcards. Quizlet. <https://quizlet.com/984638886/software-engineering-study-guide-flash-cards/?funnelUUID=035f453d-bc10-4ac0-9592-62e6b12d78e8>
- Quizlet, Inc. (2025). La investigación científica detrás de cómo funciona Quizlet. Quizlet. <https://quizlet.com/es/features/how-quizlet-works>
- Salinas Ibáñez, J. M., & Agudelo Velásquez, O. L. (2022). Editorial del número especial: Educación flexible en la era del conocimiento abierto. Edutec: Revista electrónica de tecnología educativa(79), 1-9. <https://dialnet.unirioja.es/servlet/articulo?codigo=8404148>
- Sánchez García, J. E., Gutiérrez Herrera, B. E., & Armenta Nieblas, J. L. (2016). Propuesta de diseño instruccional de un MOOC con base en la teoría de la elaboración y el Mastery Learning. Ra Ximhai, 12(6), 437–456. <https://doi.org/10.35197/rx.12.01.e3.2016.29.js>
- Sevilla Godínez, H. (2017). Educar en la era digital. Guadalajara: Pandora.
- Sommerville, I. (2011). Ingeniería de Software (9ª edición). Educación, México.
- Sosa Díaz, M. J., & Valverde Berrocoso, J. (2022). Hacia una educación digital. Modelos de integración de las TIC en los centros educativos. Revista Mexicana de Investigación Educativa, 27(94), 939-970. <https://doi.org/S1405-66662022000300939>
- Torres Vivar, L. R., Sánchez Avila, P. d., Sabando Maldonado, K. K., & Maldonado Ríos, I. D. (2024). Análisis de los modelos de enseñanza empleados en el ámbito universitario, características, ventajas y desventajas aplicación eficaz en diferentes disciplinas. RECIMUNDO, 8(2), 443-477. [https://doi.org/10.26820/recimundo/8.\(2\).abril.2024.443-477](https://doi.org/10.26820/recimundo/8.(2).abril.2024.443-477)
- Treviño, R., & Irais, S. (30 de 12 de 2024). 20 cursos gratis en línea del Tec que puedes tomar este 2025. CONECTA-Tecnológico de Monterrey.

<https://conecta.tec.mx/es/noticias/nacional/educacion/cursos-gratis-en-linea-del-tec>

Udemy. (2010). Empecemos a aprender. Udemy. <https://www.udemy.com>

Udemy. (02 de 2025). Ingeniería de Software y Arquitectura de Software. Udemy Ingeniería de software. <https://www.udemy.com/course/ingenieria-de-software-y-principios-de-arquitectura/?srsltid=AfmBOoqLDZkKsJzuBQMLe6t0iuphPH0y7UE2QNBe9SGMBFpVWfFQfyLY&couponCode=LETSLEARNNOW>

Universidad Católica Los Ángeles de Chimbote. (s.f.). Teoría del aprendizaje de Robert Gagné. ULADECH. https://files.uladech.edu.pe/docente/41916979/PS_APRENDIZAJE/sesion_8/lectura_gagne.pdf

Yuan, L., & Powell, S. (2013). MOOCs and Open Education: Implications for Higher Education. JISC CETIS. <https://doi.org/10.13140/2.1.5072.8320>