



Universidad de las Ciencias
Informáticas

FACULTAD 4

Módulo de Solicitudes de Trámites Docentes para el Sistema de Admisión de la Universidad de Ciencias Informáticas

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

Autores: Rolando Falcon Ochoa

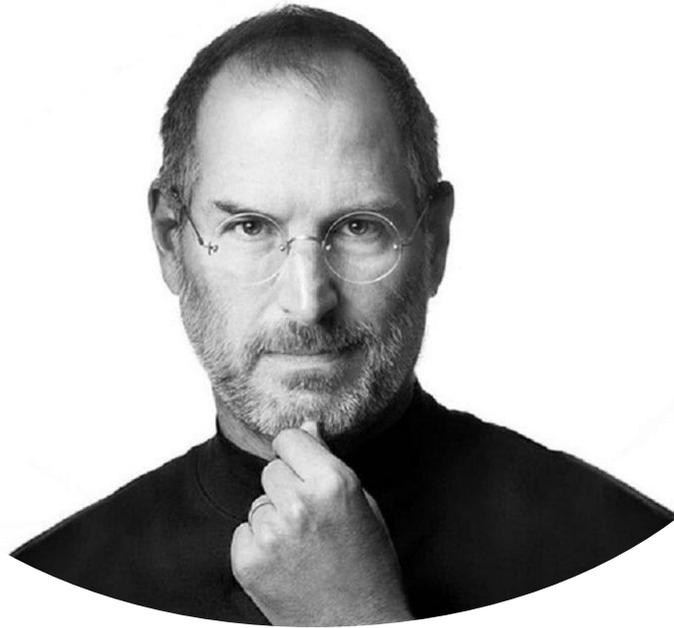
Lisette Oliva Alvarez

Tutores: M.Sc. Yadira Ramírez Rodríguez

Ing. Yosbel Falero Vento

La Habana, diciembre de 2023

Año 65 de la Revolución

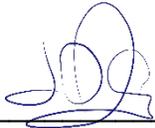


La clave no es aprender a programar, es aprender a resolver problemas. "

Steve Jobs

Declaración de autoría

Los autores del trabajo de diploma con título: ***“Módulo de Solicitudes de Trámites Docentes para el Sistema de Admisión de la Universidad de Ciencias Informáticas”*** conceden a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declaran como únicos autores de su contenido. Para que así conste firma la presente a los 8 días del mes de diciembre del año 2023.



Autor

Lisette Oliva Álvarez



Autor

Rolando Falcon Ochoa



Tutor

M.Sc. Yadira Ramírez Rodríguez



Tutor

Ing. Yosbel Falero Vento

Datos del contacto

Yadira Ramírez Rodríguez, graduada de Ingeniería en Ciencias Informáticas en el año 2007. Máster en Calidad de Software y profesora auxiliar con más de 15 de años de experiencia en la asignatura de Ingeniería de Software. Jefa del Departamento Docente de Informática de la Facultad 4 de la Universidad de las Ciencias Informáticas. Líneas de Investigación: Ingeniería de Sistemas y Calidad de Software.

Correo electrónico: yramirezr@uci.cu

Yosbel Falero Vento, graduado de Ingeniería en Ciencias Informáticas en el año 2007, director del Departamento de Informatización de la Universidad de Ciencias Informáticas con más de 15 años de experiencia en el sector informático. Líder del proyecto Picta.

Correo electrónico: yfalero@uci.cu

Dedicatoria

A mi familia, principalmente a mis padres que me lo han dado todo

Rolando Falcon Ochoa.

A mis padres.

Lisette Oliva Alvarez

Agradecimientos

Rolando Falcon Ochoa.

Quiero expresar mi más profundo agradecimiento a todas las personas que han hecho posible este proyecto, que es el resultado de muchos años de esfuerzo, dedicación y aprendizaje.

Agradezco a mis padres, a mis hermanos y a toda mi familia, que son el pilar de mi vida, por brindarme su amor incondicional, su confianza plena y su apoyo constante. Ellos han sido mi luz, mi ejemplo y mi motivación. Sin ellos, nada de esto hubiera sido posible. Gracias por acompañarme siempre, por formarme como persona y por sacrificarse por mí.

También agradezco a mi pareja Dariannis, que es el regalo más maravilloso que he recibido, por llenar mi vida de amor, comprensión y colaboración. Gracias por caminar a mi lado, por compartir mis sueños y por hacerme feliz.

Además, agradezco a mis tutores, el profesor Yosbel y la profesora Yadira, por guiarme con su orientación, sus consejos y sus correcciones. Gracias por confiar en mí, por enseñarme y por exigirme.

Asimismo, agradezco de manera especial al profesor Ramón, por su asesoría, su experiencia y su generosidad. Gracias por ayudarme, por compartir su conocimiento y por facilitarme los recursos.

Igualmente, agradezco especialmente a mi compañera de tesis; Lisette, por aportar su trabajo, su compromiso y su creatividad. Hemos formado un equipo dinámico y hemos superado juntos los obstáculos y los desafíos. Gracias por tu confianza, tu apoyo y tu amistad.

Por último, agradezco a mis amigos, por darme su ánimo, su ayuda, su alegría y su compañía. Ellos han sido mi fuente de energía, de diversión y de consuelo. Gracias por escucharme, por animarme y por celebrar conmigo cada avance y cada logro.

Finalmente, agradezco a todos los que de alguna manera han contribuido a la realización de este proyecto, aunque no los haya mencionado. Les estoy muy agradecido de todo corazón.

Lisette Oliva Alvarez.

A mis padres por apoyarme y guiarme siempre, por ser mis ejemplos a seguir, por confiar en mí. Porque sin ellos no hubiera llegado hasta aquí y por ser los mejores del mundo.

A mis abuelos, tíos, suegros y a toda mi familia, por su cariño y apoyo a lo largo de mi vida.

A mi novio por estar siempre y apoyarme incondicionalmente, por su tiempo y su paciencia.

A mi mejor amiga, por ser mi ejemplo y confidente.

A la Universidad de Ciencias Informáticas por haberme hecho la persona que soy hoy. A mi facultad y a mis profesores de toda la carrera, por su tiempo y enseñanzas.

A todos los amigos que me ha dado la universidad, que durante estos años han sido parte de mi vida y mi aprendizaje. En especial a mi amigo y compañero de tesis, por su apoyo y comprensión, por su paciencia en los momentos difíciles.

A mis tutores por su apoyo, su tiempo y sus orientaciones, sin ellos no hubiese llegado hasta aquí.

Gracias a todas las personas que de una forma u otra me han apoyado, ayudado, aconsejado, animado durante toda mi carrera.

Resumen

Las solicitudes de trámites docentes experimentan una transformación significativa con la prevalencia de plataformas digitales y sistemas en línea, optimizando las fases de solicitud, revisión, aprobación y legalización de documentos académicos. La adopción de tecnologías modernas ha resultado en una gestión más eficiente, aliviando la carga administrativa y mejorando la accesibilidad para estudiantes y personal educativo. Con el propósito de perfeccionar este proceso, la investigación se centró en la implementación de un módulo de solicitudes de trámites docentes desarrollado con Drupal 9 e integrado al Sistema de Admisión de la Universidad de Ciencias Informáticas. La metodología XP se usa asegurar una adaptación dinámica a los cambios y garantizar la satisfacción de los usuarios finales. La aplicación de exhaustivas pruebas de integración y aceptación fue para validar la funcionalidad del módulo en el contexto de solicitud de trámites docentes, proporcionando conocimientos valiosos que contribuyeron de manera significativa a la mejora continua del proceso.

Palabras clave: módulo de trámites docentes, plataformas digitales, metodología xp, pruebas de integración.

Abstract

The application of educational procedures has undergone a significant transformation with the prevalence of digital platforms and online systems, optimizing the application, review, approval and legalization phases of academic documents. The adoption of modern technologies has resulted in a more efficient management, easing the administrative burden and improving accessibility for students and educational staff. In order to improve this process, the research focused on the implementation of an educational application module developed with Drupal 9 and integrated with the Admission System of the University of Computer Science. The XP methodology is used to ensure dynamic adaptation to changes and guarantee end-user satisfaction. Extensive integration and acceptance testing were applied to validate the functionality of the module in the context of the application process, providing valuable insights that contributed significantly to the continuous improvement of the process.

Keywords: teaching procedures module, digital platforms, xp methodology, integration tests.

Índice

Introducción.....	1
Capítulo 1. Fundamentación teórica para el módulo de solicitudes de trámites docentes.	6
Introducción.....	6
1.1 Caracterización de los procesos de solicitudes de trámites docentes.	6
1.2 Análisis de plataformas que gestionan solicitudes de trámites docentes.	7
1.2.1 Plataforma Trámites a Distancia.	8
1.2.2 Certificación Académica online.....	9
1.2.3 UniLegal. Plataforma de Legalización.	10
1.2.4 Sistema de Gestión Académica Xauce AKADEMOS.....	11
1.2.5 Comparación de sistemas homólogos.....	14
1.3 Sistemas para la integración y la gestión de las solicitudes de trámites docentes	15
1.4 Metodologías de desarrollo de software.	16
1.5 Tecnologías y herramientas asociadas al desarrollo.	21
1.5.1 Herramientas.	24
1.5.2 Sistema de gestión de bases de datos (SGBD).....	27
1.5.3 Lenguaje de programación.....	28
1.5.4 Maquetado y estilos.....	30
Conclusiones del capítulo.....	32
Capítulo 2 Análisis y diseño del módulo para solicitudes de trámites docentes.	33
Introducción.....	33
2.1 Descripción de la propuesta de solución.....	33
2.2. Requerimientos no funcionales	35
2.3 Fase I. Planificación	37
2.3.1 Historias de Usuarios	37
2.3.2 Estimación de esfuerzo por Historia de Usuario	41
2.4 Fase II: Diseño	43
2.4.1 Representación arquitectónica.....	43
2.4.2 Patrones de diseño	45
2.4.3 Tarjetas CRC	49

2.4.4 Modelo de despliegue	50
Conclusiones del capítulo.....	51
Capítulo 3. Implementación y prueba del módulo para solicitudes de trámites docentes.....	53
Introducción.....	53
3.1 Fase III: Desarrollo.....	53
3.1.2 Tareas de ingeniería.	53
Tareas de ingeniería para la Iteración I.	53
Tareas de ingeniería para la Iteración II	55
3.1.3 Estándares de codificación	56
3.2 Fase IV: Pruebas	6
3.2.1 Pruebas de funcionalidad.....	8
3.2.1.2 Análisis de las pruebas de funcionalidad.....	9
3.2.2 Pruebas de integración	10
3.2.2.1 Análisis de las pruebas de integración	11
3.2.3 Pruebas de aceptación.....	13
3.2.3.1 Análisis de las pruebas de aceptación.....	17
Conclusiones del capítulo.....	18
Conclusiones.....	19
Recomendaciones.....	20
Referencias Bibliográficas	21
Anexos.....	27

Índice de tablas.

Tabla 1. Trámites docentes. Fuente: elaboración propia	7
Tabla 2. Comparación de sistemas homólogos. Fuente: elaboración propia.....	14
Tabla 3. Comparación de metodologías ágiles. Tomado de (Meléndez et al., 2019), (Scrum, 2023), (Agile Lite, 2019), (Parada et al., 2019).	18
Tabla 4. Usuarios del sistema. Fuente: elaboración propia.....	35
Tabla 5. Historia de usuario 1. Fuente: elaboración propia.	38
Tabla 6. Historia de usuario 2. Fuente: elaboración propia.	38
Tabla 7. Historia de usuario 3. Fuente: elaboración propia.	39
Tabla 8. Historia de usuario 4. Fuente: elaboración propia.	40
Tabla 9. Historia de usuario 17. Fuente: elaboración propia.....	40
Tabla 10. Historia de usuario 18. Fuente: elaboración propia.....	40
Tabla 11. Puntos de estimación. Fuente: elaboración propia.	41
Tabla 12. Plan de duración de las iteraciones. Fuente: elaboración propia.....	42
Tabla 13. Plan de entrega de versiones. Fuente: elaboración propia.	43
Tabla 14. Tarjeta CRC 1. Fuente: elaboración propia.	50
Tabla 15. Tarjeta CRC 5. Fuente: elaboración propia.	50
Tabla 16. Tarjeta CRC 6. Fuente: elaboración propia.	50
Tabla 17. Tarea de ingeniería1. Iteración 1. Fuente: elaboración propia.....	54
Tabla 18. Tarea de ingeniería 2. Iteración 1. Fuente: elaboración propia.....	54
Tabla 19. Tarea ingeniería 3. Iteración 1. Fuente: elaboración propia.	55
Tabla 20. Tarea de ingeniería 4. Iteración 1. Fuente: elaboración propia.....	55
Tabla 21. Tarea de ingeniería 5. Iteración 2. Fuente: elaboración propia.....	55
Tabla 22. Tarea de ingeniería 6. Iteración 2. Fuente: elaboración propia.....	56
Tabla 23. Pruebas. Fuente: elaboración Tomado de (Díaz et al., 2020).	7
Tabla 24. Prueba de integración con el CMS Drupal usando el gestor de base de datos PostgreSQL. Fuente: elaboración propia.	10
Tabla 25. Prueba de aceptación 1. Iteración 1. Fuente: elaboración propia.....	13
Tabla 26. Prueba de aceptación 2. Iteración 1. Fuente: elaboración propia.....	14
Tabla 27. Prueba de aceptación 3. Iteración 1. Fuente: elaboración propia.....	14
Tabla 28. Prueba de aceptación 4. Iteración 1. Fuente: elaboración propia.....	15

Tabla 29. Prueba de aceptación 5. Iteración 2. Fuente: elaboración propia.....	16
Tabla 30. Prueba de aceptación 6. Iteración 2. fuente: elaboración propia.....	16
Tabla 31. Historia de Usuario 5. Fuente: elaboración propia.	27
Tabla 32. Historia de usuario 6. Fuente: elaboración propia.....	27
Tabla 33. Historia de usuario 7. Fuente elaboración propia.	28
Tabla 34. Historia de usuario 8. Fuente: elaboración propia.....	29
Tabla 35. Historia de usuario 9. Fuente: elaboración propia.....	29
Tabla 36. Historia de usuario 10. Fuente: elaboración propia.....	30
Tabla 37. Historia de usuario 11. Fuente: elaboración propia.....	31
Tabla 38. Historia de usuario 12. Fuente: elaboración propia.....	31
Tabla 39. Historia de usuario 13. Fuente: elaboración propia.....	31
Tabla 40. Historia de usuario 14. Fuente: elaboración propia.....	32
Tabla 41. Historia de usuario 15. Fuente: elaboración propia.....	33
Tabla 42. Historia de usuario 16. Fuente: elaboración propia.....	33
Tabla 43. Historia de usuario 18. Fuente: elaboración propia.....	34
Tabla 44. Tarjeta CRC 2. Fuente: elaboración propia.	35
Tabla 45. Tarjeta CRC 3. Fuente: elaboración propia.	35
Tabla 46. Tarjeta CRC 4. Fuente: elaboración propia.	35

Índice de figuras

Figura 1. Representación de arquitectura de Drupal 9. Tomado de (Drupal, 2023a).	22
Figura 2. Propuesta de solución. Fuente: elaboración propia.	34
Figura 3. Representación arquitectónica. Fuente: elaboración propia.	44
Figura 4. Patrón controlador. Fuente: elaboración propia.	46
Figura 5. Patrón creador. Fuente: elaboración propia.	46
Figura 6. Patrón alta cohesión. Fuente: elaboración propia.	47
Figura 7. Patrón observador. Fuente: elaboración propia.	48
Figura 8. Patrón fábrica. Fuente: elaboración propia.	48
Figura 9. Patrón comando. Fuente: elaboración propia.	49
Figura 10. Patrón estrategia. Fuente: elaboración propia.	49
Figura 11. Diagrama de despliegue. Fuente: elaboración propia.	51
Figura 12. Estructuras de control. Fuente: elaboración propia.	2
Figura 13. Estructura de control. Fuente: elaboración propia.	2
Figura 14. Etiquetas y comentarios. Fuente: elaboración propia.	3
Figura 15. Uso de comillas. Uso de punto y coma. Operadores. Funciones. Fuente: elaboración propia.	4
Figura 16. Nombre y descripción de los módulos. Fuente: elaboración propia.	4
Figura 17. Rutas. Nombre de clases y métodos. Requisitos de permisos. Opciones de ruta. Fuente: elaboración propia.	5
Figura 18. Nombre y descripción de los módulos. Fuente: elaboración propia.	6
Figura 19. Prueba de integración. Fuente: elaboración propia.	12
Figura 20. Prueba de integración. Fuente: elaboración propia.	12
Figura 21. Resultado de las pruebas de integración. Fuente: elaboración propia.	12
Figura 22. Módulo de Trámites. Fuente: elaboración propia.	13
Figura 23. Prueba unitaria testEnviarDatosValido (). Fuente: elaboración propia.	36
Figura 24. Prueba unitaria función de validación {% if logueado and es_uci %}. Fuente: elaboración propia.	36
Figura 25. Resultados de pruebas funcionales. Configuración de envío de correos. Fuente: elaboración propia.	37

Índice de gráficas

Gráfica 1. Análisis de pruebas funcionales. Fuente: Elaboración propia.....	10
Gráfica 2. Análisis de pruebas de aceptación. Fuente: Elaboración propia.	17

La Habana, 17 de noviembre de 2023

“Año 65 de la Revolución”

Dirección de Informatización de la Universidad de las Ciencias Informáticas.

Acta de Conformidad

Por este medio se comunica la satisfacción de la Dirección de Informatización de la Universidad de las Ciencias Informáticas con el Módulo de Solicitudes de Trámites Docentes para el Sistema de Admisión de la Universidad de Ciencias Informáticas entregado por Rolando Falcon Ochoa y Lisette Oliva Alvarez. Se certifica que el producto cumplió con los 19 requisitos especificados y fue entregado en los plazos establecidos. El sistema fue probado y cumple con los estándares requeridos. Este módulo será utilizado en el sistema actual con el fin de contribuir al progreso de la universidad. Por ello, el centro acepta el software en su estado actual y se compromete con la utilización del mismo en los términos y condiciones acordados, así como guiar la continuidad del desarrollo de las recomendaciones realizadas.

Cordialmente,



Ing. Yosbel Falero Vento

Director de Informatización

Universidad de las Ciencias Informáticas

Introducción

La introducción del uso de las Tecnologías de la Información y las Comunicaciones (TICs), incluida la Internet, ha impactado en la configuración y progreso de las sociedades contemporáneas. Su impronta ha marcado de diferentes maneras a todos los sectores, convirtiendo la digitalización de sus procesos en la base de la era digital (Ortí, 2015). Las TICs han contribuido a impulsar la innovación en casi todas las áreas de trabajo. Han facilitado la transformación y la optimización de la mayoría de los procesos administrativos, la salud y la educación. Al mismo tiempo, han propiciado una incipiente transformación del sistema organizativo, productivo, competitivo y en el desarrollo científico técnico.

Desde su llegada a los escenarios nacionales se han convertido en un elemento indispensable para establecer nuevas líneas en el desarrollo económico, político y social del país. La estrategia para insertar el uso de las TICs es conocida como el proceso de informatización de la sociedad cubana. Pasos fundamentales de este proceso fueron la creación de la Red Nacional de los Joven Club de Computación y Electrónica en el año 1987, de INFOMED, en 1992 y de la Universidad de las Ciencias Informáticas (UCI), en el año 2002. Esta última nació en el marco del “Proyecto Futuro” implementado por el gobierno cubano con el objetivo fundamental de la formación de recursos humanos en este campo (Carbó, 2016).

Desde su fundación la UCI ha sido una de las universidades cubanas referentes en la calidad de su formación y transferencia de los resultados de sus investigaciones, con un alto componente tecnológico. La innovación tecnológica y su aplicación son parte esencial de su misión y su estrategia. Por ello, dentro del nuevo contexto de desarrollo tecnológico, se asumen nuevos preceptos encaminados a la transformación digital. Se integran las tecnologías digitales en todos los ámbitos de la sociedad, en los que las personas son el centro del hacer. Por ello, explotar las capacidades dinámicas, la formación calificada y transferencia tecnológica para adaptar sus propuestas de valor, según las necesidades de los agentes involucrados son nuevos enfoques en el encadenamiento de los procesos productivos.

Como parte de estos procesos, la UCI ha desarrollado distintas herramientas y plataformas digitales para potenciar transformaciones en el trabajo docente e institucional. Se destacan, entre ellas, la página principal de la universidad y el Sistema de Admisión (UCI, 2021). La

primera constituye el portal informativo que brinda al mundo la actualidad universitaria, procesos y programas que se desarrollan en la institución. El segundo, ofrece distintas opciones para gestionar la actividad académica de posgrado con alcance nacional e internacional y abarca desde cursos hasta la formación doctoral.

Entre las herramientas para el control docente dispone del Sistema de Gestión Académica XAUCE AKADEMOS (UCI, 2023). Esto contribuye al perfeccionamiento de los procesos académicos de la institución. Su uso permite el desarrollo coherente de una estrategia organizacional que articula a todos los niveles de decisión en los procesos universitarios. De ahí que estas herramientas y plataformas han permitido reducir la presencialidad de las actividades docentes e introducir nuevas formas de gestión académica, administrativa y estructural de la universidad. Aumentando así la eficiencia, productividad y calidad de la estructura orgánica de la institución, específicamente en la secretaría general de la universidad.

La secretaría general es la unidad organizativa de dirección de los procesos de registro general, control de la documentación, expedición y legalización de certificaciones y títulos, así como la orientación y control metodológico del trabajo que se desarrolla en las secretarías docentes de sus facultades. Es el punto de contacto principal entre la universidad y otras entidades e instituciones, tanto nacional como internacionalmente (Resolución No. 80/22, 2022). Se encarga de mantener actualizada la dirección universitaria. Facilita la información necesaria para el análisis relacionado con el ingreso, el movimiento de la matrícula, los resultados académicos, el egreso y posgrado, así como, los trámites docentes una vez finalizados los estudios universitarios de pre y postgrado (Resolución No. 80/22, 2022).

Los trámites docentes son uno de los procesos más importantes durante y tras culminar la formación básica y posgraduada. Aunque su manejo difiere en todo el mundo; es imprescindible su procesamiento, control, custodia y conservación. En algunos países como Estonia, España y Argentina los trámites docentes se realizan de forma electrónica a través de plataformas digitales, lo que permite mayor eficiencia y agilidad en su gestión. No obstante, los métodos tradicionales son aún los más utilizados. De este modo, se incrementa el riesgo de errores en el registro y archivo de la documentación, así como de su deterioro. Igualmente

requiere mayor espacio físico de almacenamiento y aumenta el consumo de tiempo en su gestión de recursos humanos y materiales.

Actualmente en la UCI, para la gestión de estos trámites docentes se requiere total presencialidad por parte de los solicitantes. Específicamente, los trámites referidos a la expedición de documentos para el posgrado son tramitados en la secretaría general de la UCI mediante AKADEMOS. A pesar de tener este sistema automatizado aún se requiere durante todo el proceso de solicitud y entrega la intervención de los usuarios autorizados. Estos hechos afectan principalmente a los egresados que no pertenecen a la provincia donde radica la universidad. Lo que trae consigo mayor inversión de tiempo y recursos de las partes involucradas. Asimismo, los solicitantes carecen de un medio para constatar el estado de su solicitud, pues únicamente se cuenta con la vía telefónica y en casos muy específicos el correo electrónico (personal) para dicho servicio. Por demás, este método resulta mucho más propenso a errores humanos y, por consiguiente, induce a la tardanza en los tiempos de respuesta a los solicitantes.

Por tanto, se identifica la necesidad de reducir la presencialidad de los procesos administrativos y propiciar una gestión más eficiente. En este sentido, las experiencias durante la pandemia de COVID-19 para la gestión docente y académica, muestran la posibilidad de hallar soluciones que transformen el actual proceso de gestión de trámites de gestión de documentos en la institución.

De acuerdo con la situación anteriormente descrita, se plantea **problema de investigación**:
¿Cómo agilizar el proceso para solicitud de trámites docentes en la Universidad de Ciencias Informáticas?

A partir de ahí, se determina como **objeto de estudio**: sistemas de información para la gestión universitaria.

Se define como **objetivo general**: desarrollar un módulo de solicitud de trámites docentes para el Sistema de Admisión de la UCI; enmarcado en el **campo de acción**: módulo de trámites docentes.

Para dar cumplimiento al objetivo general propuesto, se trazaron las siguientes **tareas de investigación**:

1. Análisis de los fundamentos teóricos para la comprensión de la gestión de trámites docentes.
2. Identificación de las funcionalidades con las que debe cumplir la solución para cumplir con su objetivo.
3. Diseño de una solución para conectar el Sistema de Admisión con el módulo archivo de AKADEMOS.
4. Implementación de la solución para la gestión de solicitudes de trámites docentes.
5. Realización de pruebas para la validación del módulo para minimizar fallos y asegurar la calidad requerida del sistema, antes de la puesta en funcionamiento.

Para el logro del objetivo, se utilizarán los siguientes **métodos de investigación científica**:

Métodos teóricos:

- Análisis y síntesis: Para la comprensión del entorno en el que se desarrolla el proceso de gestión de solicitud de trámites en la Universidad de Ciencias Informáticas. Permitirá identificar los contenidos relacionados con el proceso de gestión de solicitud de trámites desde entornos web y el diseño de la propuesta de solución.
- Análisis documental: Para abordar los aspectos conceptuales y metodológicos de la investigación.

Técnicas e instrumentos para la recogida de datos

- Entrevista: Se realiza al personal de la secretaría general (técnico y directivo) para la recopilación exhaustiva de los requisitos del módulo.

La estructura del reporte consta de una introducción, el desarrollo compuesto por tres capítulos, conclusiones, recomendaciones y el listado de referencias bibliográficas que fueron utilizadas durante la investigación. Además, posee también una serie de anexos.

Capítulo 1. Fundamentación teórica del módulo para solicitudes de trámites docentes. En él se presentan los elementos conceptuales que sustentan la situación problemática planteada. Se muestran las herramientas y tecnologías identificadas para el desarrollo web. Se analizan los sistemas de gestión de información y se definen los lenguajes de programación y las metodologías para el desarrollo web.

Capítulo 2. Análisis y diseño del módulo para de solicitudes de trámites docentes. Explica el proceso de desarrollo de software propuesto para la solución. Se exponen los artefactos a tener en cuenta para ello. De acuerdo a la metodología seleccionada XP, se exponen los requisitos funcionales y no funcionales, arquitectura del software y el modelo de datos. También se definen los patrones de diseño y de arquitectura a utilizar durante el desarrollo de la solución.

Capítulo 3. Implementación y prueba del módulo para solicitudes de trámites docentes. Aborda aspectos relacionados con la implementación del sistema a desarrollar. Se realizan pruebas de software para la validación y aceptación de la solución propuesta, así como, la comprobación de las funcionalidades y las especificaciones requeridas.

Capítulo 1. Fundamentación teórica para el módulo de solicitudes de trámites docentes.

Introducción

En el presente capítulo se analizan las bases conceptuales y características relevantes en el desarrollo del módulo web para la gestión de solicitudes de trámites docentes. Se presenta un estudio de sistemas con características similares al módulo que se propone desarrollar y se describen los sistemas implicados en el proceso. Se define el uso de la metodología de desarrollo de software a utilizar, las herramientas y tecnologías con vistas a que el nuevo módulo cumpla con los requisitos del cliente y sea compatible con el Sistema de Admisión y el Sistema de Gestión Académica Xauce AKADEMOS.

1.1 Caracterización de los procesos de solicitudes de trámites docentes.

La solicitud, expedición y legalización de la documentación docente es un proceso sumamente importante en el entorno de la gestión académica. Estos documentos evidencian la veracidad de planes de formación, evaluaciones y titulaciones de los estudiantes. Toda persona graduada tiene el derecho de solicitarlos, siempre que estén avalados por una causa probada. Esta gestión de documentación académica se denomina trámites docentes (Resolución No. 80/22, 2022).

En el contexto educacional y según las normas que se establecen en el país se entiende como trámite docente a la gestión de los procesos relacionados con actividades de registro general, documentación, expedición y legalización de certificaciones en las instituciones de Educación Superior. Se realiza a través de la secretaría general y las secretarías docentes, tanto para las carreras y programas de Nivel de Educación Superior de Ciclo Corto que se cursan en el pregrado, así como para los diferentes programas de posgrado. (Resolución No. 80/22, 2022) Por tanto, basado en la información recopilada puede decirse que los procesos de trámites docentes son aquellos que permiten gestionar y documentar las actividades académicas de estudiantes y egresados en las instituciones de Educación Superior. Estos procesos involucran diferentes pasos y diligencias que deben cumplirse para garantizar la calidad y transparencia en todo momento.

Estos trámites se realizan con vistas a su uso en el exterior o el territorio nacional (Resolución No. 80/22, 2022). Se clasifican según el nivel de formación académica en dos tipos: de pregrado y posgrado como se muestra en la tabla 1. Estos son:

Tabla 1. Trámites docentes. Fuente: elaboración propia

Trámites docentes de pregrado	Trámites docentes de posgrado
<ul style="list-style-type: none"> ● registro de matrícula de nuevo ingreso; ● ratificación de matrícula para estudiantes continuantes; ● solicitudes para repitencia; ● procedimientos con expedientes; ● control de asignaturas; ● control de arrastres entre otros. 	<ul style="list-style-type: none"> ● certificación de estudios universitarios terminados; ● certificación de estudios parciales; ● plan temático de asignaturas; ● copia literal certificada de títulos y diplomas; (validación de fotocopias del título) ● certificado de índice académico (certificación de notas); ● certificado de subsanación de error o emisión en nombre o apellido; ● certificación de cualquier otra información solicitada por el graduado y que aparezca reflejada en su expediente académico; ● y otras certificaciones de carácter académico o docente relacionadas con el pregrado o la Educación de Posgrado y que aparezca reflejada en el expediente académico.

Con la aplicación de nuevas tecnologías a la gestión de trámites docentes en las universidades se ha producido una mutación de los procesos manuales tradicionales hacia su digitalización. Diversas experiencias han sido implementadas para su optimización, la reducción del tiempo de espera y los errores humanos. Para fundamentarlas, se presenta en el siguiente epígrafe, un estudio comparativo de diferentes soluciones aplicadas a la gestión de trámites.

1.2 Análisis de plataformas que gestionan solicitudes de trámites docentes.

Para la realización del estudio de soluciones homólogas existentes se tuvieron en cuenta una serie de criterios planteados y aprobados por el cliente, estos son:

- Dominio de aplicación.

- Registro y autenticación de usuarios.
- Verificación de usuarios.
- Notificaciones.
- Firma digital.
- Servicio de código QR

Algunas de las soluciones homólogas encontradas son:

1.2.1 Plataforma Trámites a Distancia.

La plataforma Trámites a Distancia de la Universidad de Buenos Aires (TAD-UBA) es un sistema de inicio de trámites a distancia (TAD) en línea que reemplaza el trámite tradicional en papel por ventanilla. Simplifica la presentación, seguimiento y transparencia. Esta plataforma permite también el avance del trámite y si se presentan dificultades, se informa a sus usuarios (TAD-UBA, 2023).

Etapas (SIET, 2023):

- Facultad
- Envío a Rectorado
- Envío a Facultad

Se puede hacer a través del sitio del TAD UBA en la primera etapa y a través del Sistema Integrado de Expedición de Títulos (SIET) en la última etapa (SIET, 2023).

Cuando en el SIET dice “Diploma Terminado” o aparece la leyenda «Este sistema solo muestra trámites en curso. Una vez concluidos solo es posible consultarlos por un lapso de 60 días». Se podrán retirar sin cita previa todos los días lunes de 10 a 17hs en el sitio que figura en la página, con documento de identidad nacional. Si la persona solicitante no puede concurrir, puede ser retirado por un representante, siempre que, al hacerlo, cuente con un poder ante escribano o un apoderamiento gestionado a través del sistema TAD Argentina que avale dicha autorización (TAD-UBA, 2023).

Tipos de trámites: (TAD-UBA, 2023)

- Certificado analítico.
- Certificado de Promedio (incluye a plazos si los hubiese).

- Plan de estudios integral (plan de Estudios completo que integra el plan de estudios, su carga horaria, duración en años de la carrera y la escala de calificaciones en el mismo documento).
- Legalizaciones de certificados (constituyen un trámite por separado de la emisión del certificado).
- Duplicados de diplomas por robo, hurto o destrucción.
- Diploma de Honor.

1.2.2 Certificación Académica online.

La Universidad de Castilla-La Mancha (UCLM) pone a disposición de sus estudiantes, a través de su Sede Electrónica, un procedimiento on-line para la tramitación y obtención de certificaciones académicas personales de los estudios cursados. La certificación tramitada a través de la Sede de la UCLM, permite a la persona interesada obtener un documento firmado electrónicamente con plena validez legal y de forma completamente automatizada. Esto aplica tanto en lo relativo a la solicitud como al abono de los precios públicos correspondientes, mediante sistema de pago con tarjeta de crédito. Todos estos servicios se realizan a través de la red en la secretaría virtual (UCLM, 2023).

Acceso

- El acceso a esta gestión será exclusivamente por Certificado digital o Usuario y contraseña (activa) del estudiante.

Consideraciones

- Las certificaciones académicas se generan de forma automática con la información contenida en las bases de datos de la UCLM en el momento en que se solicita dicho certificado. Debe asegurarse que la información recogida emite un borrador para verificar los datos, previo a la emisión.
- Actualmente solo están habilitados para este procedimiento un modelo de Certificación personal básico y un modelo de asignaturas superadas para estudiantes titulados, en función de cada uno de los subtipos de estudios. En caso de necesitar una certificación diferente, su emisión Unidad de Gestión Académica de Campus se encarga de hacerlo por el procedimiento habitual.

Seguimiento de expedientes

- Cuando la persona interesada ha finalizado una solicitud o la ha dejado pendiente de finalizar, debe acceder a su “Área Personal” dentro de la Sede. De este modo, en caso de entrar de nuevo en catálogo de servicios, haría una nueva solicitud, que lógicamente no procede, pues sería innecesario.
- El acceso al Área Personal permite ver los expedientes tramitados y descargar los documentos asociados a cada caso (certificados o justificantes de pago).

1.2.3 UniLegal. Plataforma de Legalización.

UniLegal es una plataforma desarrollada por Departamento de Desarrollo de Software, Dirección de Informatización, de la Universidad Máximo Gómez Báez de Ciego de Ávila que permite la legalización online de sus títulos universitarios cubanos. Con la nueva herramienta se elimina el requisito de la presencialidad para la tramitación y acorta los tiempos de un proceso que antes podía tardar varios días (UNICA, 2022).

Entre las operaciones que maneja la plataforma se encuentran el registro de usuario, crear y verificar solicitudes de pregrado, visualizar y descargar documentos. Además, permite mediante la opción de reportar problemas, emitir quejas o sugerencias con respecto al funcionamiento del sitio. Todos estos procedimientos se ejecutan a través de formularios.

Funcionamiento de UniLegal

- Para realizar las operaciones en el sitio, deberá crearse primeramente un usuario. Este paso requiere la introducción de datos personales: nombre y apellidos, correo electrónico, número del carné de identidad y un nombre de usuario y contraseña.
- Una vez creado el usuario se procede a autenticarse para acceder a realizar las operaciones del sitio. (Fotocopia del título universitario).

Procedimientos de validación

- La aplicación comprueba que la persona verdaderamente es graduada de la Universidad de Ciego de Ávila, mediante la búsqueda en los registros. Luego confirma a terceros que ese egresado es realmente quien dice ser.

- La aplicación genera un código de barras y un QR. Al escanearlo, remite a una página donde aparecen los datos específicos del usuario en cuestión; y a la cual puede acceder cualquier persona para verificar que ese documento sea verdadero.
- UniLegal cuenta con 400 usuarios registrados pertenecientes a las provincias de Artemisa, Mayabeque y casi todas las de la región oriental.

1.2.4 Sistema de Gestión Académica Xauce AKADEMOS.

El Sistema de Gestión Académica (SGA) XAUCE AKADEMOS tiene como objetivo gestionar la información de los procesos académicos para contribuir a la toma de decisiones a partir del registro, control y seguimiento del tránsito del estudiante por los subsistemas del Sistema Nacional de Educación que atiende el MINED. Es una herramienta multiplataforma que contribuye al perfeccionamiento de los procesos académicos de una institución (*XAUCE AKADEMOS 1.0 | Universidad de las Ciencias Informáticas, 2023*).

El uso de esta herramienta permite el desarrollo coherente de una estrategia organizacional que articule todos los niveles de decisión presentes en los procesos universitarios. Todos los roles del proceso educativo están involucrados en la solución. Este sistema posibilita el acceso a la información de forma segura a todos los niveles y brinda acceso tanto a estudiantes como a docentes. Está compuesto por varios módulos, entre los que se halla el módulo de solicitud de trámites docentes. En este se gestionan todos los documentos de solicitudes realizadas por parte de entidades o egresados (*XAUCE AKADEMOS 1.0 | Universidad de las Ciencias Informáticas, 2023*).

Entre los documentos más solicitados para la gestión de trámites docentes de posgrado se encuentran la validación de fotocopias del título, la certificación de notas, el certificado de estudios universitarios terminados y el certificado de subsanación de errores. Acorde a la Resolución No.80/22 el tiempo de respuesta, los requerimientos y procedimientos de validación previstos en la gestión de trámites docentes se ajustan a las siguientes disposiciones:

Tiempo de respuesta

- La secretaría general dispone de hasta cuarenta y cinco (45) días hábiles para confeccionar y entregar el documento solicitado, según dispone el Acuerdo del Comité Ejecutivo del Consejo de Ministros de fecha 4 de marzo de 2014. (Artículo 391)

Requerimientos y/o procedimientos de validación

- Cuando en la secretaría general sea necesario expedir un Certificado enmendando o subsanando la omisión o un error de nombre y/o apellidos, u otro dato en el título universitario, emitido por dicho centro de estudios. El solicitante entrega una fotocopia de su carné de identidad, que se archiva en el expediente académico junto con la copia del certificado.
- Excepcionalmente se exigirá la entrega de una certificación de nacimiento del titular de dicho título universitario (Artículo 439.1).

Requerimientos

- El trámite se realiza en el Archivo Central o la secretaría general, según corresponda, a partir de la solicitud del tramitador o persona autorizada al trámite representante de la entidad de servicios jurídicos autorizada. También, por solicitud de la persona a quien se le expidió el título, siempre que presente documento oficial de una autoridad competente del organismo, entidad o dependencia, requiriendo la comprobación y confirmando su uso y destino para fines académicos o laborales.
- Junto con la solicitud se puede presentar el título original o no. No es obligatorio presentar fotocopia para validar la autenticidad del título.
- Cuando no se presenta el título el solicitante debe entregar todos los datos que permitan verificar su expedición. Los datos son: nombre y apellidos del titular; carrera o programa de nivel superior de ciclo corto; fecha de expedición; y los datos del asentamiento en los registros de graduados de la Facultad y la secretaría general (Tomo, Folio y Número).
- En circunstancias necesarias se solicitará, si fuese posible, presentar el título original para verificar las firmas de las autoridades académicas.
- Otra forma de legalizar las fotocopias de los títulos es poner en su reverso un cuño confirmando que el título es válido. (Artículo 443).
- El certificado de estudios terminados se expide en cualquier momento posterior a la graduación, confeccionado en la secretaría general o Archivo Central, para usarla con fines laborales o académicos legalmente establecidos. (Artículo 400-406).

En las condiciones actuales para la gestión de trámites docentes de posgrado en la UCI se combinan métodos tradicionales que requieren de la presencialidad de la persona solicitante y la presentación de los documentos para su validación con la digitalización de procesos a través del sistema AKADEMOS.

Para dar continuidad al proceso de solicitud se realizó una entrevista a la secretaria del archivo y secretaría general respectivamente. Según la información recopilada se describe la ejecución real del proceso de solicitud y expedición de documentos para egresados de la UCI.

Según las partes implicadas se ha establecido un mecanismo de priorización de las solicitudes de trámites que se reciben en este orden:

- Las entidades empresariales o bufetes colectivos.
- Solicitudes personales de la persona egresada o a través de un familiar.

Además, cada solicitud es recepcionada y procesada en el Archivo General. Posteriormente, se envía a secretaría general donde es firmada y cuñada para su entrega entre 15 y 30 días hábiles.

Pasos y requerimientos para expedir:

- Certificación de fotocopias del título: cada solicitante debe presentar un total de tres copias en físico de sus documentos. Posteriormente, se verifica que el tomo, folio y número que consta en su fotocopia de certificación coincida con los que aparecen en el sistema. Si es veraz la información, entonces se procede a dar curso a la solicitud.
- Certificación de notas se verifica que el índice académico que consta en la certificación de notas original coincida con el que aparece en el sistema. Si es veraz la información, entonces se procede a dar curso a la solicitud.
- Para la subsanación de errores, el solicitante debe presentar el número de serie de su carné de identidad y una carta que sustente porqué solicita este procedimiento. Una vez comprobada la veracidad de la información se procede a dar curso a la solicitud.
- Certificado de estudios universitarios terminados el solicitante debe presentar un documento que avale el motivo de su solicitud. Posteriormente se procede a dar curso a la solicitud.

En caso de situaciones donde no coincide la información entre los documentos archivados en la secretaría docente, la presentada por las personas solicitantes y la registrada en el sistema

se adopta como válida la que consta en el sistema. En él se encuentra archivada la información del expediente académico. Estos errores implican una extensión en el tiempo de respuesta, en tanto se requiere una inversión de tiempo mayor en la validación de los documentos.

1.2.5 Comparación de sistemas homólogos.

Considerando las características de los sistemas presentados, teniendo en cuenta la realización del proceso en la universidad y los criterios seleccionados se establece la siguiente comparación:

Tabla 2. Comparación de sistemas homólogos. Fuente: elaboración propia

Criterios	Trámites a Distancia	Certificación Académica on-line	UniLegal	Módulo de trámites docentes AKADEMOS
Dominio de aplicación	Web	Web	Web	Desktop
Notificaciones	Vía correo y seguimiento en línea	Vía correo o postal	Vía correo	No
Registro y autenticación de usuarios	Para el registro se introducen los datos: <ul style="list-style-type: none"> • Nombre y apellidos • CUIT/CUIL (clave fiscal) • Correo electrónico • Teléfono 	Para el registro de introducen los datos del solicitante: <ul style="list-style-type: none"> • Introducir certificado digital • Correo electrónico • Teléfono • Nombre de usuario 	Para el registro de introducen los datos: <ul style="list-style-type: none"> • Nombre y apellidos • Correo electrónico • Número del carné de identidad • Nombre de usuario y contraseña. 	No
Verificación de usuarios	Sí	Sí	Sí	Sí
Firma digital	Junto a los documentos requeridos se adjunta la firma digital.	Junto a los documentos requeridos se adjunta la firma digital.	No	No
Servicio de código QR	Sí	Sí	Sí	No

En la tabla anterior se establece la comparación entre los sistemas homólogos seleccionados. Se tienen en cuenta los criterios de comparación: dominio de aplicación, notificaciones, registro y autenticación de usuarios, verificación de usuarios, firma digital, servicio de código QR. Estos aspectos son indispensables para establecer una similitud con el módulo de solicitudes de trámites docentes a desarrollar. Pues en ellos se evidencian las carencias o beneficios con los que cuenta dicho proceso.

Luego del análisis realizado se percibe como generalidad que todas las plataformas seleccionadas tienden a usar la web como la solución más óptima a la hora de solicitar trámites docentes. Además, se toma como referente el criterio de notificaciones que utilizan las plataformas de Trámites a Distancia y UniLegal. De este modo, a través del correo electrónico se enviarán automáticamente las notificaciones a los usuarios implicados. Además, se podrá visualizar el estado de las solicitudes en tiempo real desde la página. Asimismo, algunas de las características obligatorias que deben tener este tipo de soluciones son: registro, autenticación y verificación de usuarios.

Por tanto, para lograr que las solicitudes de trámites docentes que se solicitan y gestionan desde la secretaría general y en AKADEMOS tengan un mayor alcance para todos los egresados se hace necesario utilizar alguno de los sistemas con los que cuenta la universidad.

1.3 Sistemas para la integración y la gestión de las solicitudes de trámites docentes

La UCI cuenta actualmente con un grupo de servicios informáticos en los que incluye programas académicos de pregrado y posgrado, herramientas para la educación a distancia, así como otros servicios de alta calidad y profesionalidad, disponibles para personas naturales y jurídicas tanto nacionales como extranjeras. Para los programas académicos de posgrado se cuenta con el Sistema de Admisión un sitio web que brinda un conjunto de prestaciones que podrían adaptarse en función de las necesidades descritas al inicio de la investigación (UCI, 2023).

El Sistema de Admisión, desarrollado en Drupal, posee un diseño intuitivo y amigable, que facilita la navegación y la búsqueda de información. Cuenta con medidas de seguridad para proteger los datos de los usuarios y garantizar la transparencia del proceso. Asimismo, dispone de funcionalidades como el registro, la autenticación y verificación de usuarios necesarias para brindar un servicio de calidad a los usuarios. Por todo lo anteriormente mencionado, el Sistema

de Admisión, será la plataforma donde se alojarán los servicios para realizar la solicitud de trámites docentes.

Aun así, estos procesos deben seguir cumpliendo con las normativas establecidas y siendo evaluadas por la secretaría general. De ahí que, sea imprescindible establecer una sincronización directa con el SGA Xauce AKADEMOS ya que este continuará siendo el escenario donde se acrediten todos estos procedimientos.

Para el desarrollo de las solicitudes de trámites docentes se hace necesario el uso de herramientas y tecnologías como editores de texto, entornos de desarrollo local, gestores de base de datos, así como metodologías de desarrollo de software y patrones de diseño. A la hora de elegir las se tienen multitud de opciones y la selección de las actas deberá estar encaminada al tipo de aplicación que se desea desarrollar, el tiempo que se posee y la comodidad del desarrollador. Este epígrafe trata sobre las herramientas, tecnologías y metodología de desarrollo de software. A continuación, se describen las características de cada una de ellas.

1.4 Metodologías de desarrollo de software.

Las metodologías de desarrollo de software son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de software informático. Las diversas metodologías de desarrollo de software se pueden clasificar en dos grandes grupos: las tradicionales y las ágiles. (Meléndez et al., 2019)

Las metodologías tradicionales, como el modelo en cascada, se basan en un proceso secuencial y rígido. En ellas se definen los requisitos al inicio del proyecto y se realizan las fases de análisis, diseño, implementación, pruebas y mantenimiento de forma lineal. Estas metodologías tienen la ventaja de ser formales y documentadas. Sin embargo, presentan inconvenientes como la poca adaptabilidad al cambio, la baja participación del cliente y el alto riesgo de fracaso.

Las metodologías ágiles están más centradas en proyectos pequeños, en el factor humano, en la colaboración con el cliente y en el desarrollo incremental de software con iteraciones muy cortas. Es un enfoque que ha mostrado ser efectivo ante requisitos cambiantes y cuando se requiere reducir el tiempo de desarrollo al tiempo que se mantiene una alta calidad. (Parada et al., 2019) (Canós et al., 2018).

Las metodologías ágiles tienen otros beneficios como la simplificación de la sobrecarga de los procesos, mejora en la previsibilidad con una adecuada gestión del riesgo, mejor perfil de productividad, entre otros (Parada et al., 2019).

Debido a las diferencias notables entre el desarrollo de software tradicional y el ágil, los requisitos dinámicos y el modelo de desarrollo evolutivo de módulo. Además, del tamaño del equipo, el tiempo de duración, la participación del cliente desde el comienzo y las revisiones se realizan luego de cada iteración. Se decide que la solución siga un enfoque ágil.

Dentro del mundo de las metodologías ágiles, se puede encontrar un amplio abanico de posibilidades de aplicación, entre ellas destacan:

- *Extreme Programming* (XP, por sus siglas en inglés) Programación Extrema: se caracteriza por los ciclos de desarrollo cortos, el incremento de los planes para el desarrollo del software, además de la retroalimentación que se establece con el cliente (Meléndez et al., 2019).
- **SCRUM**: describe la forma en que los miembros de equipo deben trabajar para poder obtener un sistema flexible en un entorno que varía de manera constante (Scrum, 2023). El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.
- *Agile Lite*: “(...)puede ser aplicada a cualquier proyecto, asumiendo que el trabajo a realizar se pueda dividir en pequeñas acciones.” (Agile Lite, 2019). Utiliza ciclos de desarrollos cortos.
- *Crystal Method*: son un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas (Parada et al., 2019).
- *Feature-Driven Development 9* (FDD). Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño

e implementación del sistema partiendo de una lista de características que debe reunir el software (Parada et al., 2019).

Después de caracterizar estas metodologías y teniendo en cuenta los siguientes parámetros se establece la siguiente comparación:

Tabla 3. Comparación de metodologías ágiles. Tomado de (Meléndez et al., 2019), (Scrum, 2023), (Agile Lite, 2019), (Parada et al., 2019).

Aspectos	XP	SCRUM	Agile Lite	Crystal Method	Feature-Driven Development 9
Duración	Suelen trabajar en iteraciones que son de una o dos semanas de duración. En ocasiones pueden ser más.	Suelen trabajar en iteraciones (llamadas sprints) que son a partir de dos semanas a un mes de duración.	Ciclos de desarrollo cortos	-	Las iteraciones son cortas.
Cambios	En XP son mucho más sensibles al cambio dentro de sus iteraciones.	Los equipos de Scrum no permiten cambios en sus 'sprints'.	División del trabajo en pequeñas acciones.	Entregas frecuentes.	-
Prioridad	Suelen trabajar en un orden de prioridad estricta y se desarrollarán las que son priorizadas por el cliente	Prioriza la acumulación de productos, pero el equipo determina la secuencia en la que se desarrollarán los elementos.	-	-	Se centra en las fases de diseño e implementación del sistema.

Prácticas de ingeniería	Prescribe prácticas de ingeniería, las cosas se van desarrollando en base a pruebas, el enfoque en las pruebas automatizadas, la programación en parejas, diseño simple, refactorización, etcétera.	No prescribe ninguna práctica de ingeniería.	-	Fácil acceso a usuarios expertos que permite al equipo obtener requerimientos de primera mano y retroalimentación inmediata.	Satisfacer los requerimientos del usuario, los aspectos técnicos deben ser excelentes, independientemente de su cantidad y complejidad.
Tamaño del proyecto	Pequeños y medianos.	Pequeños, medianos y grandes.	Pequeños, medianos y grandes.	Pequeños.	Pequeños.
Tamaño de equipo	Menor que 10.	Múltiples equipos menores que 10.	-	Equipos pequeños y equipos grandes que desarrollan sistemas de baja y alta criticidad.	Equipos pequeños y equipos grandes que desarrollan sistemas de baja y alta criticidad.

Teniendo en cuenta los criterios que aparecen en la tabla anterior y que durante el desarrollo del módulo las iteraciones no deben ser cortas, que el proyecto es pequeño, el tamaño del equipo es de dos integrantes y son necesarias las prácticas de ingeniería para poder documentar todo el proceso de desarrollo de la solución, se determina entonces como metodología de desarrollo de software seleccionar XP (*Extreme Programming*). XP reconoce que los proyectos tienen que trabajar para lograr una reducción de costos y explotar los ahorros una vez que se han ganado. Entre sus principales características figuran:

Artefactos de la metodología XP

- Historia de usuarios: Las historias de usuarios (HU) representan una breve descripción del comportamiento del sistema. Se realizan por cada característica principal del sistema y se utilizan para cumplir estimaciones de tiempo y el plan de lanzamientos. También reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación. Cada HU debe ser lo suficientemente comprensible y delimitada

para que los programadores puedan implementarlas en unas semanas (Meléndez et al., 2019).

- Tareas de ingeniería: Una HU se descompone en varias tareas de ingeniería, que describen las actividades que se realizarán en cada HU, así mismo las tareas de ingeniería se vinculan más al desarrollador, ya que permite tener un acercamiento con el código (Meléndez et al., 2019).
- Tarjetas CRC: Las tarjetas Clase\Responsabilidades\Colaboradores permiten conocer qué clases componen el sistema y cuales interactúan entre sí. (Meléndez et al., 2019)

Fases de la programación XP

- Planificación: La planificación como un diálogo continuo entre las partes involucradas en el proyecto. Esto incluye al cliente, a los programadores y a los coordinadores. El proyecto comienza recopilando las historias de usuarios, las que constituyen los tradicionales casos de uso. Una vez obtenidas estas historias de usuarios, los programadores evalúan rápidamente el tiempo de desarrollo de cada una y determinan así, el tiempo total de desarrollo del software (Joskowicz, 2015).
- Diseño: Especificación de cómo debe ser el diseño final de la aplicación. Enfatiza en la facilidad de implementación de un diseño sencillo con respecto a uno complejo (Joskowicz, 2015).
- Desarrollo: Implementación de los códigos necesarios para satisfacer las historias de usuario.
- Pruebas: Una vez terminada la codificación, se deben realizar las pruebas pertinentes a la misma (Joskowicz, 2015).

La metodología de desarrollo de software que se escoge para el desarrollo de la solución es XP, debido a que se trata de una metodología ágil que ofrece ventajas como la rapidez, la flexibilidad y la calidad en el desarrollo de aplicaciones web. Además, XP se adapta al contexto del proyecto, ya que el equipo de desarrollo es pequeño, se requiere seguir las prácticas de ingeniería, está sujeta a cambios según las necesidades del cliente y se dispone de poco tiempo para la ejecución. Por estas razones, se considera que XP es la metodología más apropiada para el desarrollo del módulo de solicitudes de trámites docentes.

1.5 Tecnologías y herramientas asociadas al desarrollo.

Dado que el módulo para solicitud de trámites docentes está integrado al Sistema de Admisión y hace uso de algunas de sus funcionalidades, se hace obligatorio adoptar la tecnología y herramientas en las que este está desarrollado. Además, de ser un requisito del cliente. Con esto se garantiza el cien por ciento de la compatibilidad a la hora de la integración del módulo. Por ello se utiliza:

Sistema de Gestión de Contenido (CMS)

Con esta denominación se conoce la herramienta de software que permite crear, organizar y publicar documentos y otros contenidos de forma colaborativa. Los sistemas de gestión de contenidos (Content Management System, en adelante CMS) están formados por un conjunto de aplicaciones web que, de un modo similar a un portal, operan tanto en Internet como en una intranet (González et al., 2021).

Su principal ventaja consiste en el hecho de que permiten organizar y mostrar contenidos sin que sea necesario poseer grandes conocimientos de programación web. Con su ayuda se pueden generar y publicar noticias, crear taxonomías (sistemas de clasificación) con las que el administrador clasifica contenidos. También se pueden insertar logotipos e imágenes personalizadas o corporativas del portal, añadir secciones, administrar las bases de datos de usuarios, entre otras (González et al., 2021).

Drupal 9

Drupal es un software de gestión de contenido. Se utiliza para crear muchos de los sitios web y aplicaciones que se usan con frecuencia. Drupal tiene excelentes características estándar, entre las que se reconocen la creación de contenido sencilla, rendimiento confiable y excelente seguridad. Se distingue, además, por su flexibilidad. La modularidad es uno de sus principios fundamentales. Sus herramientas le ayudan a crear el contenido estructurado y versátil que resulta necesario en las experiencias web dinámicas (Drupal, 2023).

También es una excelente opción para crear marcos digitales integrados. Puede ampliarse con uno o múltiples complementos. La funcionalidad de Drupal puede ampliarse a través de los módulos. Los temas le permiten personalizar la presentación de su contenido. Las

distribuciones son paquetes de Drupal empaquetados que se pueden usar como kits de inicio. Es posible mezclar y combinar estos componentes para mejorar las habilidades centrales de esta herramienta. Igualmente, se puede integrar Drupal con servicios externos y otras aplicaciones en su infraestructura. Se considera que ningún otro software de gestión de contenido es tan potente y escalable.

El proyecto Drupal es un software de código abierto. Cualquiera puede descargarlo, usarlo, trabajar en él y compartirlo con otros. Se basa en principios como la colaboración, la globalización y la innovación. Se distribuye bajo los términos de la Licencia Pública General GNU (GPL). No se requiere abonar una tarifa de licencia para su uso, por lo que resulta accesible (Drupal, 2023).

Drupal trabaja con una arquitectura de cinco capas: datos, módulos, bloques y menús, permisos de usuario y plantillas como se muestra en la figura 1.

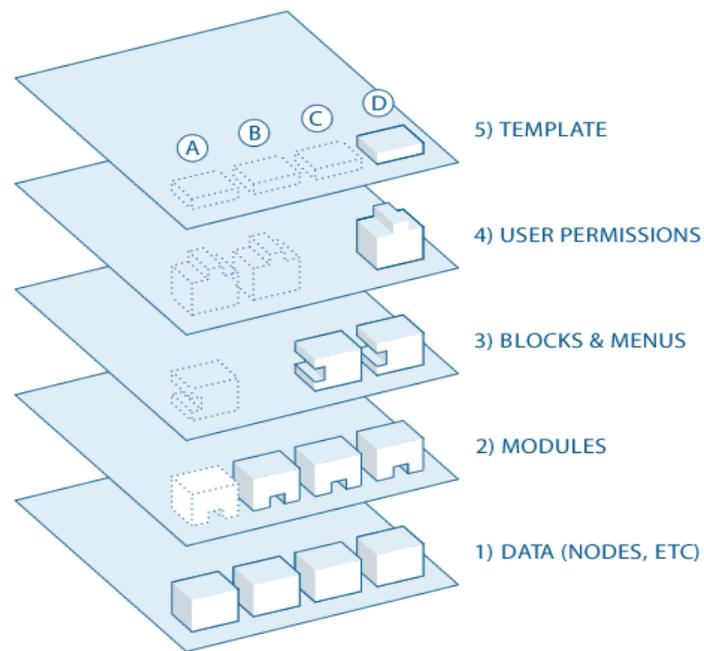


Figura 1. Representación de arquitectura de Drupal 9. Tomado de (Drupal, 2023a).

A continuación, la descripción de cada una de las capas de la arquitectura de Drupal.

- Datos (nodos, etc.): en la base del sistema está la colección de nodos, el conjunto de datos. Antes de que algo pueda mostrarse en el sitio, debe introducirse como datos.
- Módulos: la siguiente capa es donde se alojan los módulos. Los módulos son plugins funcionales que forman parte del núcleo de Drupal (se entregan con Drupal) o son elementos contribuidos que han sido creados por miembros de la comunidad Drupal. Los módulos se basan en la funcionalidad del núcleo de Drupal, lo que permite personalizar los elementos de datos (campos) en sus tipos de nodo; configurar el comercio electrónico; ordenar y mostrar contenido mediante programación (salida personalizada controlada por los filtros que defina); y mucho más. Existen miles de opciones diferentes en el creciente repositorio de módulos Drupal contribuidos. Representan la innovación y el esfuerzo colaborativo de todo el mundo, desde individuos a grandes corporaciones.
- Bloques y menús: en la siguiente capa, se encuentran bloques y menús. Los bloques a menudo proporcionan la salida de un módulo o pueden ser creados para mostrar lo que se quiere y luego pueden ser colocados en varios lugares (regiones) en el diseño de tu plantilla (tema). Los bloques pueden configurarse para que se muestran de varias maneras, así como para que sólo se muestran en ciertas páginas definidas, o sólo para ciertos usuarios definidos. Los menús son navegadores en Drupal, que definen el contenido que viene en cada ruta de menú definida (URL relativa). Los menús son un elemento central de Drupal que proporcionan enlaces a todas las páginas creadas en Drupal.
- Permisos de usuarios: a continuación, están los permisos de usuario. Aquí es donde se configuran los ajustes para determinar lo que los diferentes tipos de usuarios se les permite hacer y ver. Se definen permisos para varios roles y, a su vez, se asignan usuarios a estos roles para concederles los permisos definidos.
- Plantillas (temas): en la capa superior está el tema del sitio (la "piel"). Esto se compone principalmente de HTML5 y CSS, con algunas variables Twig entremezclados, por lo que el contenido generado por Drupal puede ir en los lugares apropiados. También se incluye con cada tema un conjunto de funciones que se pueden utilizar para anular las

funciones estándar en los módulos con el fin de proporcionar un control completo sobre cómo los módulos generan su marcado en el momento de la salida. Las plantillas también se pueden asignar sobre la marcha en función de los permisos de usuario.

1.5.1 Herramientas.

Las selecciones de las herramientas específicas desempeñan un papel crucial en el proceso de generación de horarios. Estas herramientas proporcionan funcionalidades y características que facilitan la manipulación de datos, la implementación de algoritmos y la visualización de resultados.

Herramienta CASE y lenguaje para el modelado.

Una herramienta CASE es conjunto de programas especializados y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un software. La inclusión de las herramientas CASE están tomando cada vez más relevancia en la planeación y ejecución de proyectos que involucran sistemas informáticos, debido a que permiten aumentar la productividad en el desarrollo de software. Estas herramientas, suelen inducir a sus usuarios a la correcta utilización de metodologías que implican una reducción tanto en el costo del proyecto, como en el tiempo de desarrollo del producto de software final (Battaglia et al., 2017).

Visual Paradigm 8.0.

Es una potente herramienta profesional CASE (por sus siglas en inglés Computer Aided Software Engineering) que utiliza el modelado UML para el desarrollo de aplicaciones. Soporta el ciclo completo del desarrollo del software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue, permite dibujar todos los tipos de diagramas de clases y generar documentación. Por lo que es ideal para ingenieros de software, analistas y arquitectos de sistemas que están interesados en construir sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. Se considera muy completa y fácil de usar, con soporte multiplataforma y excelentes facilidades de interoperabilidad con otras aplicaciones (Visual Paradigm, 2021).

En esta investigación se decide utilizar esta herramienta en su versión 16.2, por las ventajas que ofrece. Ya que genera código y permite exportarlo en HTML, tiene licencia gratuita y comercial, utiliza un lenguaje estándar común y su diseño es centrado en casos de uso y enfocado en el negocio lo que permite generar un software de mayor calidad.

Lenguaje de modelado UML.

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es un lenguaje de modelado visual común para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. Este lenguaje describe, a través de diagramas, los límites, la estructura y el comportamiento de un sistema. Es el más conocido y utilizado en la actualidad por ser el mejor para visualizar, especificar, documentar y construir los métodos o procesos de un sistema (Larman, 2005).

UML consiste en diferentes tipos de diagramas que describen al sistema y los objetos que contiene, este representa buenas prácticas para la construcción y documentación de diferentes aspectos del modelado de sistemas de software y de negocios. Está compuesto por diversos elementos gráficos que se combinan, y existen reglas para combinar tales elementos. Lo que hace posible representar diversas perspectivas de un mismo sistema. Por lo tanto, este será el lenguaje de modelado utilizado durante la investigación.

Visual Studio Code Versión 17.2.

Visual Studio Code es un editor de código fuente ligero, pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para *JavaScript*, *TypeScript* y *Node.js* y tiene un rico ecosistema de extensiones para otros lenguajes (como C++, C#, Java, Python, PHP, Go) y tiempos de ejecución (Visual Studio Code, 2022).

La depuración suele ser la característica que más extrañan los desarrolladores en una experiencia de codificación más eficiente, así que se hizo posible. Visual Studio Code incluye un depurador interactivo, por lo que puede recorrer el código fuente, inspeccionar variables, ver pilas de llamadas y ejecutar comandos en la consola. Visual Studio Code también se

integra con herramientas de compilación y secuencias de comandos para realizar tareas comunes y acelerar los flujos de trabajo diarios. *VS Code* es compatible con *Git*, por lo que puede trabajar con control de código fuente sin salir del editor, incluida la visualización de diferencias de cambios pendientes (Visual Studio Code, 2022).

XAMPP 3.3.0.

XAMPP es una abreviatura en la que la *X* significa "*Cross-Platform*" (plataforma cruzada), la *A* significa "*Apache*", la *M* significa "*MYSQL*" y la *P* significa "*PHP*" y "*Perl*", respectivamente, es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar (Marquez, 2022).

XAMPP proporciona un entorno de desarrollo local ideal, con la intención de lograr hacer que el alojamiento de aplicaciones *PHP* creadas con *XAMPP* sea lo más fácil posible. Entre las principales ventajas de las características del servidor *XAMPP* se puede destacar que la mayor ventaja que tiene sobre cualquier otro servidor web es que es fácil de configurar y usar, además es un software multiplataforma disponible para todo tipo de sistemas operativos como Linux y Windows. También posee muchos otros módulos esenciales como *phpMyAdmin*, *OpenSSL*, *MediaWiki*, *WordPress*, *Joomla* y algunos más. Por otra parte, se puede encontrar como desventaja principal que la configuración es muy difícil de realizar en comparación con otros programas de similar funcionamiento, como es el caso de *Wamp* (XAMP, 2023).

GitLab 16.4.1.

GitLab es una plataforma de *DevSecOps* integral y de código abierto que se ejecuta en su servidor o en la nube. Viene con soporte incorporado para el control de versiones, seguimiento de problemas, revisión de código, CI/CD y más. GitLab tiene un rico ecosistema de integraciones para otros lenguajes y tiempos de ejecución (GitLab, 2023).

La integración continua (CI) y la entrega continua (CD) son características que los desarrolladores aprecian en una experiencia de codificación eficiente, y GitLab las ha hecho posibles. GitLab incluye un sistema de CI/CD interactivo, por lo que puede automatizar sus pipelines, inspeccionar variables de entorno, ver trabajos y etapas, y ejecutar comandos en la

consola. GitLab también se integra con herramientas de compilación y secuencias de comandos para realizar tareas comunes y acelerar los flujos de trabajo diarios (GitLab, 2023).

GitLab es compatible con Git, por lo que puede trabajar con el control de código fuente sin salir de la plataforma, incluyendo la visualización de diferencias de cambios pendientes (GitLab, 2023).

PgAdmin 4-7.7.

PgAdmin es la herramienta oficial para administrar bases de datos PostgreSQL. Proporciona una interfaz gráfica intuitiva y completa que permite realizar tareas de administración, cómo crear y modificar bases de datos, ejecutar consultas SQL, administrar usuarios y configurar la replicación. PgAdmin 4 permite acceder a todas las funcionalidades de la base de datos, consulta, manipulación y gestión de datos (Sahoo, 2023).

Una de las características destacadas de pgAdmin es su herramienta de consultas, que permite ejecutar comandos SQL y analizar los resultados de forma visual. Además, puede realizar respaldos, restaurar la base de datos y ejecutar tareas de mantenimiento de forma sencilla, incluso con usuarios accediendo a los datos en modo de lectura (Sahoo, 2023).

1.5.2 Sistema de gestión de bases de datos (SGBD).

Un sistema gestor de base de datos (SGBD) o *Database Management System* (DBMS) reúne el conjunto de programas empleados para administrar y gestionar la información presente en una base de datos. Estos sirven de interfaz entre la base de datos, las aplicaciones y el usuario, permitiendo recuperar la información de forma sencilla, rápida y eficiente. Algunas de sus funciones son permitir la definición y almacenamiento de la información, acceder a los datos y modificarlos, garantizando la seguridad e integridad de los mismos. Además, se usan para efectuar análisis y generar informes, de la forma más eficiente posible. Para funcionar correctamente, los gestores de bases de datos están compuestos por un lenguaje de definición de datos, uno de manipulación y otro de consulta (UNIR, 2023a).

PostgreSQL es un sistema de bases de datos de código abierto que se destaca por su capacidad de adaptación y su cumplimiento con el estándar SQL. Actualmente se considera uno de los motores de base de datos más avanzados en la industria. El proyecto es gratuito y

libre. Además, ofrece una gran cantidad de opciones avanzadas para gestionar bases de datos relacionales en programación. PostgreSQL es una poderosa herramienta para gestionar grandes volúmenes de datos y la aplican sectores como gobierno, industria, educación y comercio (Dony, 2023).

Funciones principales

- Almacenamiento eficiente de datos: organiza y almacena datos garantizando su integridad y consistencia.
- Consultas avanzadas: permite realizar consultas complejas y analizar datos para obtener información valiosa con lenguaje SQL.
- Backend confiable para aplicaciones: empleado principalmente en el desarrollo de aplicaciones web y móviles, pues ofrece un almacenamiento seguro y escalable para los datos de la aplicación.

Ventajas destacadas de PostgreSQL

- Escalabilidad y rendimiento.
- Flexibilidad y extensibilidad.
- Comunidad activa y soporte.
- Seguridad y confiabilidad.

1.5.3 Lenguaje de programación.

Un lenguaje de programación es un elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis. Para controlar el comportamiento de una máquina, particularmente una computadora (Parra, 2018).

JavaScript.

JavaScript es uno de los recursos que han surgido para incorporar dinamismo y capacidades al lenguaje HTML. Actualmente es la tecnología más extendida en el enriquecimiento de páginas web del lado del cliente. Es un lenguaje de programación basado en prototipos, multiparadigma, dinámico y con soporte para programación orientado a objetos. Se utiliza principalmente para permitir mejoras en la interfaz de usuario y en las páginas web dinámicas.

Entre las acciones típicas que se pueden realizar en JavaScript se tienen dos vertientes. Por un lado, los efectos especiales sobre páginas web y por el otro, JavaScript permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se pueden crear páginas interactivas con programas como calculadoras, agendas o tablas de cálculo (Parra, 2018).

El lenguaje JavaScript se inserta en documentos HTML, de forma que su código queda reflejado en la propia página y no es llamado o cargado de ninguna fuente externa. Se trata de un lenguaje interpretado puro y sensible a mayúsculas, aunque algunas implementaciones ignoran en parte este último extremo. Se escoge JavaScript para el desarrollo de la presente solución informática porque proporciona los medios para controlar las ventanas del navegador y el contenido que muestran, programar páginas dinámicas simples y optimizar los tiempos de carga y el tráfico del servidor. Además, permite capturar los eventos generados por el usuario y responder a ellos, comprobar los datos que el usuario introduce en un formulario antes de enviarlos y comunicarse con dicho usuario mediante diversos métodos.

PHP 8.1.

HyperText Pre-Processor, por su acrónimo en inglés, es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. PHP es muy simple de utilizar para los principiantes, pero a su vez ofrece muchas características avanzadas para los programadores profesionales. Además, ha sido ampliamente portado y puede ser desplegado en la mayoría de los servidores web en casi todos los sistemas operativos y plataformas, de forma gratuita (PHP, 2023).

Es similar a otros lenguajes informáticos como C o C++, rápido, de sintaxis cómoda y su sencillez contribuye a su rápido aprendizaje. Es un lenguaje interpretado, multiplataforma, se le pueden agregar extensiones fácilmente y dispone de una gran cantidad de librerías (Fernández, 2021). La meta del lenguaje es permitir rápidamente a los desarrolladores la generación de páginas dinámicas. Con PHP en su versión 7.4.1 se puede realizar el procesamiento de información en formularios, foros de discusión, entre otras.

Este lenguaje será utilizado en el desarrollo del sistema por las múltiples ventajas que posee, entre las que se encuentran:

- Posee una comunidad muy grande de desarrolladores y existe abundante documentación, tutoriales, ejemplos de código, foros etc.

- Es fácil de aprender comparado con otros lenguajes de programación.
- Capacidad de conexión con la mayoría de los manejadores de base de datos.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Es un lenguaje que incluye soporte para la programación orientada a objetos.

1.5.4 Maquetado y estilos.

La maquetación y los estilos son aspectos fundamentales para el diseño y la implementación de aplicaciones web, ya que determinan la estructura, la apariencia y la interacción de las páginas web. La maquetación se refiere a la distribución de los elementos que componen una página web, tales como cabeceras, menús, contenidos, imágenes, etc. Los estilos se refieren a las propiedades visuales de los elementos, tales como colores, fuentes, tamaños, bordes, etc. Para realizar la maquetación y los estilos de un sistema web, se pueden usar diferentes herramientas que permiten crear sistemas web atractivos, funcionales y adaptables a diferentes dispositivos y resoluciones. Tales como:

HTML 5.

HyperText Markup Language o HTML es un lenguaje informático de composición de documentos que cuenta con instrucciones especiales que son interpretadas por el navegador y como resultado se muestran textos, imágenes, vídeos y enlaces. Utilizando HTML se puede definir la apariencia de las páginas creadas de diferentes formas. Contiene una gran variedad de etiquetas que son las encargadas de mantener la estructura e indicar la semántica de un documento, convirtiéndose en un lenguaje informático para presentación con soporte de aspectos visuales (Aubry, 2019).

Es un estándar reconocido en todo el mundo, multiplataforma, soportado por muchos navegadores. Con él, se puede obtener un producto atractivo y rápido sin mucho esfuerzo por parte del desarrollador. Permite establecer enlaces entre diferentes documentos y la introducción de referencias a otras páginas por medio de enlaces de hipertexto. HTML5 es la actualización de HTML y, también es un término de mercadotecnia para agrupar las nuevas tecnologías de desarrollo de aplicaciones web: HTML5, CSS3 y nuevas capacidades de JavaScript.

Se decide utilizar HTML5 ya que presenta mejoras con respecto a las anteriores versiones de HTML. Además, incluye un nuevo conjunto de etiquetas para dar estructura a un documento HTML, abandonando el uso de tablas y etiquetas <div> para la maquetación de las páginas. En esta versión, los formularios son mejorados con una redefinición de nuevos tipos de elementos para la etiqueta input, y han sido incorporados al lenguaje nuevos atributos para los elementos de formulario (Powell, 2010).

CSS 3.

Hojas de Estilos en Cascada o por sus siglas en inglés CSS, Cascading Style Sheets, es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje marcado. Es muy utilizado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML. Es una tecnología muy utilizada por muchos sitios web para crear páginas visualmente atractivas, pues permite definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, así como la posición de cada elemento dentro de la página (Aubry, 2019).

Este lenguaje será utilizado para el desarrollo de la solución propuesta pues presenta muchas ventajas, entre las que se destacan la flexibilidad, compatibilidad, continuidad, la separación del contenido y presentación y la unificación del diseño de las páginas del sitio. Además, facilita la publicación de contenidos en múltiples formatos de presentación y brinda una gran consistencia al sitio (Powell, 2010).

SASS.

SASS es una tecnología que permite mejorar la escritura y el mantenimiento de las hojas de estilo en los proyectos web. Se trata de un lenguaje que se compila a CSS, pero que ofrece características avanzadas que no están disponibles en el lenguaje estándar (UNIR, 2023b).

Algunas de estas características son:

- Variables: permiten definir valores que se pueden reutilizar a lo largo del código, como colores, tamaños, fuentes, etc.
- Anidamiento: permiten organizar las reglas CSS de forma jerárquica, siguiendo la estructura del HTML.

- Mixins: permiten definir bloques de código que se pueden incluir en diferentes lugares, evitando la repetición y facilitando la modificación.

Balsamiq Wireframes 4.7.1.

Balsamiq Wireframes es una herramienta de diseño de interfaces de usuario que facilita la creación de bocetos digitales de baja fidelidad. Estos bocetos permiten plasmar las ideas de forma rápida y sencilla, así como compartirlas y obtener retroalimentación. Balsamiq Wireframes se basa en el concepto de wireframing, que consiste en dibujar el esquema básico de una interfaz sin entrar en detalles gráficos o funcionales. De esta forma, se puede centrar la atención en la estructura, el contenido y la interacción de la interfaz, sin distraerse con elementos secundarios (Balsamiq, 2023).

Conclusiones del capítulo

En este capítulo y a partir del estudio de aplicaciones web orientadas a las solicitudes de trámites docentes en línea, fue posible arribar a las siguientes conclusiones:

- Se identificó mediante el estudio de los sistemas homólogos que las características y funcionalidades que estarán presentes en la implementación serán: dominio de aplicación, notificaciones, registro y autenticación de usuarios, verificación de usuarios, dominio de aplicación y servicio de código QR.
- Se selecciona XP como metodología de desarrollo de software, lo que permite desarrollar practicas ingenieriles y es la q más de ajusta de acuerdo al tiempo y tamaño del equipo.
- Se adopta el modelo de trabajo al que será integrado al módulo haciendo uso de las tecnologías y herramientas requeridas por el cliente como son Drupal 9, PHP 8.1, PostgreSQL 15.1, etc.

De esta manera se está en condiciones de desarrollar el módulo de solicitudes de trámites docentes para el Sistema de Admisión, utilizando las herramientas y tecnologías seleccionadas.

Capítulo 2 Análisis y diseño del módulo para solicitudes de trámites docentes.

Introducción

En el presente capítulo se presenta una propuesta de solución para los problemas detectados en el estudio del estado del arte realizado en el Capítulo 1. Se mencionan los roles que intervienen en la solución. También se definen las HU, la planificación de entrega de versiones del producto y el diseño del sistema de acuerdo a las fases que propone XP.

2.1 Descripción de la propuesta de solución.

El módulo de solicitudes de trámites docentes permitirá a los estudiantes egresados de la Universidad de Ciencias Informáticas, independientemente de su carrera, realizar solicitudes en línea para obtener certificaciones y otros procedimientos relacionados con sus documentos académicos. Para ello, módulo de trámites docentes:

- Estará integrado al Sistema de Admisión, por lo que hará uso del registro y autenticación que posee este para garantizar la autenticidad de los usuarios.
- Estará diseñado para funcionar en conjunto al SGA Xauce AKADEMOS. Desde este se ejecutarán todos los procedimientos de aprobación, denegación y/o expedición de documentos haciendo uso de una herramienta de sincronización.
- El módulo debe permitir crear las solicitudes de trámites docentes y ofrecer la información necesaria para cada tipo de trámite.
- Mediante el perfil de usuario podrá visualizar el listado de las solicitudes que ha realizado.
- Cada usuario podrá consultar el estado de sus solicitudes en su perfil de usuario.
- El módulo permitirá enviar y recibir notificaciones vía correo electrónico sobre cualquier dificultad o estado de su trámite.
- Se notificará automáticamente a la secretaría general cada vez que se registre una solicitud en el módulo de trámites docentes.

A continuación, se presenta la modelación de la propuesta de solución, teniendo en cuenta su integración al Sistema de Admisión y su conexión directa al SGA Xauce AKADEMOS:

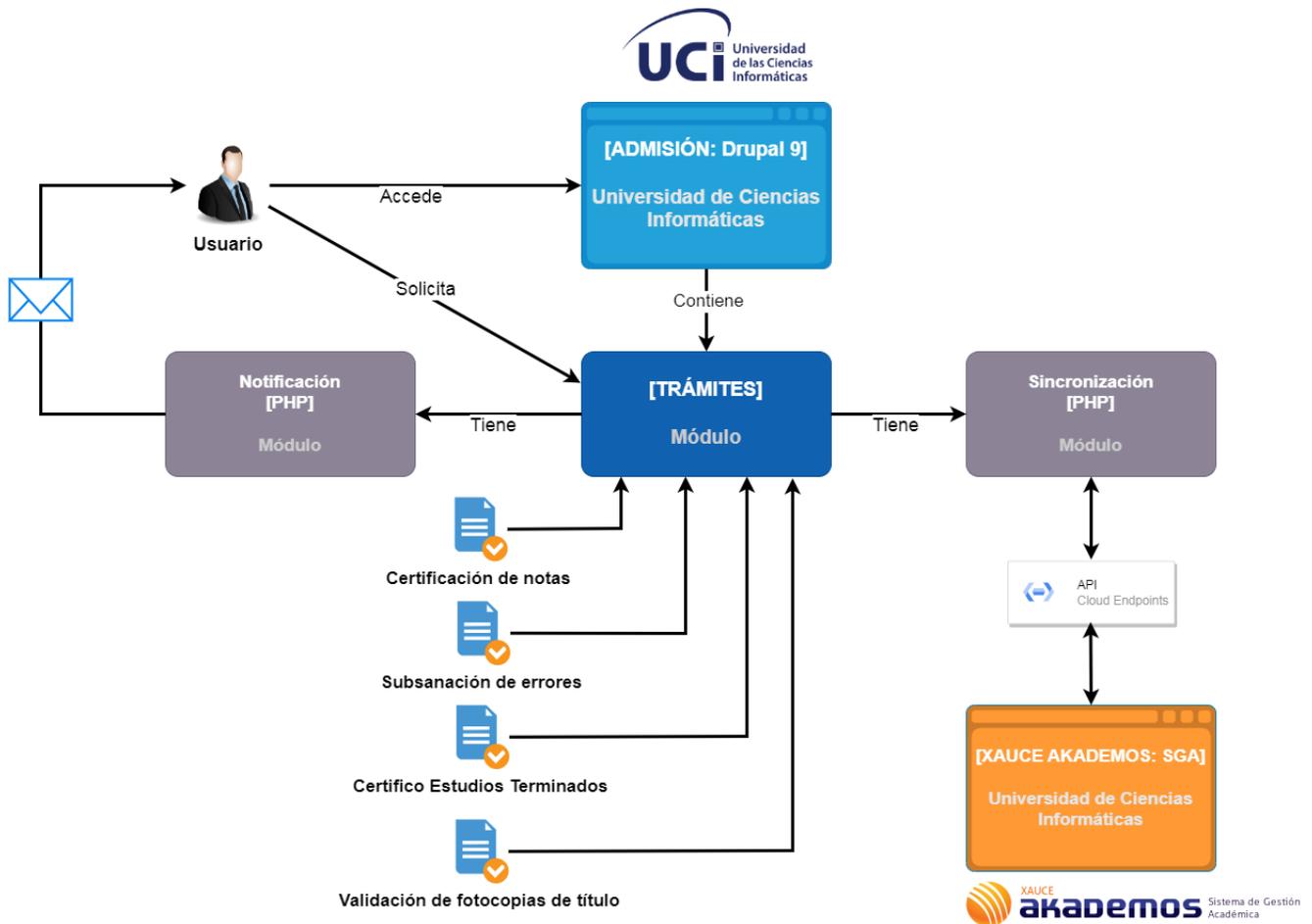


Figura 2. Propuesta de solución. Fuente: elaboración propia.

En el esquema representado en la figura 2 describe el escenario de la propuesta de solución donde el cliente (usuario) accede al Sistema de Admisión que contiene el módulo de trámites. El usuario puede solicitar diferentes tipos de trámites como son: certificación de notas, subsanación de errores, certificado de estudios universitarios terminados y validación de fotocopias del título. El módulo de trámite posee además dos módulos asociados. Uno para el envío de notificaciones y el otro para la sincronización con el SGA Xauce AKADEMOS. El primero se encarga de notificar al usuario cuando este crea una solicitud y cuando ha sido aprobada o rechazada; el segundo, se encarga de contener los métodos donde se establece la conexión con la herramienta de sincronización (API) que a su vez se conecta con el SGA Xauce AKADEMOS donde se realizan todos los procedimientos de los trámites.

Para una correcta ejecución del módulo a implementar se deben tener en cuenta los requerimientos necesarios. A continuación, se presentan los usuarios del sistema y los roles que desempeñan. Además, se describen los requerimientos no funcionales del módulo.

Tabla 4. Usuarios del sistema. Fuente: elaboración propia.

Usuarios	Roles
Administrador	Rol encargado de gestionar el sistema y gestionar usuarios. Cuenta con la mayor cantidad de permisos.
Usuario anónimo	El usuario solo tendrá acceso a la funcionalidad mostrar información.
Usuario autenticado	Rol que solo podrá visualizar información y crear los trámites.

2.2. Requerimientos no funcionales

Los Requerimientos No Funcionales (RNF), como su nombre sugiere, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa definen las restricciones y características requeridas por el software (requerimientos del producto), de la organización que desarrolla el software (requerimientos organizacionales) o de fuentes externas (Sommerville, 2005). Los requerimientos no funcionales del módulo a desarrollar son:

Requerimientos del producto:

Usabilidad

- **RNF 1:** El módulo debe mostrar la información de cada trámite legible y precisa para el usuario.

Portabilidad

- **RNF 4:** La solución se debe ejecutar en los navegadores existentes (Chrome y Mozilla) en la universidad actualmente, así como adaptar su interfaz a cualquier dispositivo que acceda al módulo.

Hardware

- **RNF 5:** El tiempo de demora de una petición al servidor debe ser menor de 2 a 5 milisegundos aproximadamente.
- **RNF 6:** Los servidores web y de base de datos deben poseer como mínimo 1 GB de memoria RAM.
- **RNF 7:** Se requiere 2 GB de memoria RAM como mínimo.
- **RNF 8:** Se requiere un procesador de 2 GHZ como mínimo.
- **RNF 9:** Se requiere un procesador de 2 GHZ como mínimo.

Requerimientos organizacionales:

Restricción en el diseño y la implementación:

- **RNF 11:** Como servidor web Apache Server con la versión PHP 8.1.17
- **RNF 12:** Como gestor de bases de datos PostgreSQL 16.0.1 con pgAdmin 4 versión 7.7.

Mantenibilidad

- **RNF 13:** Se debe permitir realizar modificaciones posteriores para mejorar el sistema o en caso de que existan nuevas exigencias por parte del cliente.
- **RNF 14:** El sistema debe estar bien documentado de manera tal que el tiempo de mantenimiento sea lo más pequeño posible.

Requerimientos externos:

Interoperabilidad

- **RNF 15:** El sistema podrá ser visualizado en dispositivos desde las resoluciones 768x1024 y 1366x768.
- **RNF 16:** Un navegador Chrome versión 83.0 o superior; o Mozilla Firefox versión 84.0 o superior.
- **RNF 17:** Sistema operativo GNU Linux Ubuntu Server 12.04 o superior o Windows 10 Home.

Seguridad

- **RNF 18:** El sistema debe gestionar y requerir información de usuarios para su uso.
- **RNF 19:** La información manejada por el módulo está protegida con el objetivo de evitar accesos no autorizados.
- **RNF 20:** Se garantizará la integridad de la información mediante mecanismos de control de acceso, utilizando usuario, contraseña y niveles de accesos para cada usuario de acuerdo a su rol, asegurando que cada uno pueda tener disponible solamente las opciones que se encuentran en correspondencia con su actividad.

En el capítulo 1 quedó seleccionado el uso de la metodología XP para guiar el proceso de desarrollo de software de la presente investigación. A continuación, se describe cómo fueron llevadas a cabo las fases de planificación y diseño. Una vez definidos e identificados los usuarios, comprendiendo sus necesidades y estableciendo los requerimientos planteados por el cliente, se iniciará la fase de planificación. Con el objetivo de crear un plan que guíe el desarrollo del proyecto de manera ágil y eficiente.

2.3 Fase I. Planificación

Es la fase en la que se define el alcance general del proyecto. En esta el cliente define lo que necesita mediante la redacción de HU y establece la prioridad de cada una. Luego, los programadores estiman los tiempos de desarrollo en base a esta información. Las estimaciones realizadas en esta fase son primarias, debido a que estarán basadas en datos de muy alto nivel y podrían variar cuando se analicen en cada iteración. Además, se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. (Joskowicz, 2015).

2.3.1 Historias de usuarios

Las HU son la técnica que utiliza XP para especificar los requisitos de software, estas deben ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a las tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra HU. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto equivale

a una semana ideal de programación (5 días laborables). Las historias, generalmente, valen de 1 a 3 puntos como máximo (esto representa las semanas) (Meléndez et al., 2019).

A continuación, se describen las HU más significativas definidas para llevar a cabo el desarrollo del módulo, las demás se contemplan en los anexos.

Tabla 5. Historia de usuario 1. Fuente: elaboración propia.

Historia de Usuario	
Número: 1	Nombre: Visualizar información de validación de fotocopias de título.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos de estimación: 0.4	Iteración asignada: 1
Descripción: El módulo debe permitir visualizar la información sobre los pasos y procedimientos para realizar la solicitud del trámite de validación de fotocopias de título.	
Observaciones: <ul style="list-style-type: none"> ● Esta información es visible para todo tipo de usuario. ● El botón de “solicitar trámite” no está disponible para usuarios que no estén autenticados. ● El botón de “solicitar trámite” no está disponible para usuarios que no son graduados UCI. 	
Interfaz: 	

Tabla 6. Historia de usuario 2. Fuente: elaboración propia.

Historia de Usuario	
Número: 2	Nombre: Crear solicitud de validación de fotocopias de título.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de estimación: 0.5	Iteración asignada: 1

Descripción: El usuario con rol usuario autenticado y graduado UCI al acceder al módulo debe visualizar el botón “solicitar trámite” al presionarlo aparecerá un formulario con los campos necesarios para crear el trámite. Debe poseer además los botones de guardar y cancelar respectivamente.

Los datos del formulario son:

- Tipo de solicitud.
- Motivo.
- Estado.
- Fecha de aprobación.
- Fecha de entrega.
- Validez.
- Fecha de solicitud.

Observaciones:

- Al guardar el formulario se envía un correo al usuario confirmando la creación de su solicitud, y la secretaria notificando la nueva solicitud.

Interfaz:

Tabla 7. Historia de usuario 3. Fuente: elaboración propia.

Historia de Usuario	
Número: 3	Nombre: Editar solicitud de validación de fotocopias de título.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de estimación: 0.3	Iteración asignada: 1
<p>Descripción: El usuario con rol de administrador accede a la opción de contenidos en el panel de administración al presionar el botón editar se debe mostrar el formulario para modificar los campos necesarios del trámite junto a los botones guardar y cancelar.</p> <p>Los datos del formulario son:</p> <ul style="list-style-type: none"> • Tipo de solicitud. • Motivo. • Validez 	
<p>Observaciones:</p> <ul style="list-style-type: none"> • El formulario está disponible solo para usuarios con rol administrador. 	

Tabla 8. Historia de usuario 4. Fuente: elaboración propia.

Historia de Usuario	
Número: 4	Nombre: Eliminar solicitud de validación de fotocopias de título.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de estimación: 0.3	Iteración asignada: 1
<p>Descripción: El usuario con rol de administrador accede a la opción de contenidos en el panel de administración, selecciona el trámite que desea eliminar y al presionar el botón de eliminar se muestra un mensaje para verificar si ese es el que se quiere eliminar. Además, se muestran los botones de eliminar y cancelar respectivamente.</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> El formulario está disponible solo para usuarios con rol administrador. 	

Tabla 9. Historia de usuario 17. Fuente: elaboración propia.

Historia de Usuario	
Número: 17	Nombre: Enviar notificación al usuario
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de estimación: 1	Iteración asignada: 2
<p>Descripción: El módulo debe permitir enviar automáticamente un correo electrónico al usuario que ha creado una solicitud de trámite docente, informando que su solicitud ha sido registrada con éxito.</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> El usuario no debe estar autenticado para recibir notificaciones. 	

Tabla 10. Historia de usuario 18. Fuente: elaboración propia.

Historia de Usuario	
Número: 18	Nombre: Enviar notificación a la secretaria.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de estimación: 1	Iteración asignada: 2
<p>Descripción: El módulo debe permitir enviar automáticamente un correo electrónico a la secretaria notificando que se ha creado una nueva solicitud desde el Sistema de Admisión.</p>	

Observaciones:

- El usuario no debe estar autenticado para recibir notificaciones.

2.3.2 Estimación de esfuerzo por Historia de usuario

Se realiza la estimación de esfuerzo que arroja cada HU, con el objetivo de obtener un correcto desarrollo del sistema. A continuación, se muestra la estimación realizada

Desarrollo del plan de iteraciones

Una vez definidas las HU y realizada una previa estimación de esfuerzos, se procede a la planificación de la etapa de implementación del sistema. En este espacio, se crea el plan de iteraciones, donde se especifica la prioridad con que se implementarán las HU organizadas por iteraciones. Teniendo en cuenta el esfuerzo asociado a las HU y a las prioridades del cliente, se define una versión que sea de valor para este.

Tabla 11. Puntos de estimación. Fuente: elaboración propia.

No.	Historias de Usuario	Puntos de estimación (semanas)
1	Visualizar información de trámite de validación de fotocopias de título.	0.4
2	Crear solicitud de validación de fotocopias de título.	0.5
3	Editar solicitud de validación de fotocopias de título.	0.3
4	Eliminar solicitud de validación de fotocopias de título.	0.3
5	Visualizar información de trámite de certificado de estudios terminados.	0.4
6	Crear solicitud de certificado de estudios terminados.	0.5
7	Editar solicitud de certificado de estudios terminados.	0.3
8	Eliminar solicitud de certificado de estudios terminados.	0.3
9	Visualizar información de trámite de certificación de notas.	0.4
10	Crear solicitud de certificación de notas.	0.5
11	Editar solicitud de certificación de notas.	0.3
12	Eliminar solicitud de certificación de notas.	0.3
13	Visualizar información de trámite de subsanación de errores.	0.4

14	Crear solicitud de subsanación de errores.	0.5
15	Editar solicitud de subsanación de errores.	0.3
16	Eliminar solicitud de subsanación de errores.	0.3
17	Enviar notificación al usuario.	1
18	Enviar notificación a la secretaría.	1
19	Visualizar solicitudes de trámites docentes.	1

Plan de duración de las iteraciones

A continuación, se presenta el plan de duración de las iteraciones. Este plan, tiene como finalidad, mostrar la duración de cada iteración, así como el orden en que serán implementadas las HU en cada iteración como se muestra en la tabla siguiente:

Tabla 12. Plan de duración de las iteraciones. Fuente: elaboración propia.

Iteración	Historias de Usuario	Duración(semanas)
1	Visualizar información de trámite de validación de fotocopias de título.	8
	Crear solicitud de validación de fotocopias de título.	
	Editar solicitud de validación de fotocopias de título.	
	Eliminar solicitud de validación de fotocopias de título.	
	Visualizar información de trámite de certificado de estudios terminados.	
	Crear solicitud de certificado de estudios terminados.	
	Editar solicitud de certificado de estudios terminados.	
	Eliminar solicitud de certificado de estudios terminados.	
	Visualizar información de trámite de certificación de notas.	
	Crear solicitud de certificación de notas.	
	Editar solicitud de certificación de notas.	
	Eliminar solicitud de certificación de notas.	
	Visualizar información de trámite de subsanación de errores.	
	Crear solicitud de subsanación de errores.	
Editar solicitud de subsanación de errores.		

	Eliminar solicitud de subsanación de errores.	
2	Enviar notificación al usuario.	3
	Enviar notificación a la secretaría.	
	Visualizar solicitudes de trámites docentes.	
Total de semanas	11	

Plan de entregas

En el plan de entrega que se presenta a continuación, se hace una propuesta de las versiones (*releases*) del sistema. Cada versión se conformará al finalizar una iteración.

Tabla 13. Plan de entrega de versiones. Fuente: elaboración propia.

Entregable	Iteración 1	Iteración 2
Módulo	Versión 1.0	Versión 2.0
Fecha	11/09/23	24/11/23

2.4 Fase II: Diseño

La fase de diseño en la metodología XP es crucial para el desarrollo exitoso de un proyecto. Durante esta etapa, se prioriza la comunicación efectiva entre el equipo de desarrollo y el cliente, lo que ayuda a evitar malentendidos y errores futuros. Se busca crear un diseño simple y claro, eliminando lo innecesario y enfocándose en las funcionalidades esenciales del sistema (Delgado, 2008).

El enfoque incremental permite adaptarse a medida que se obtiene retroalimentación del cliente, realizando ajustes en el diseño en cada iteración. Además, se realizan pruebas de diseño para asegurar la calidad del sistema y detectar posibles problemas antes de la implementación. Se busca crear un diseño flexible y adaptable, anticipando cambios futuros y permitiendo la fácil incorporación de nuevas funcionalidades (Beck & Andres, 2012).

2.4.1 Representación arquitectónica

Una arquitectura, en el contexto del desarrollo de software, se refiere a la estructura o diseño fundamental de un sistema o aplicación. La arquitectura de software proporciona una visión

global y abstracta del sistema, definiendo cómo se divide en partes, cómo se comunican entre sí y cómo se gestionan los datos y las funcionalidades (Navarro et al., 2017). Es como el plano o la estructura básica sobre la cual se construye el software.

El módulo para solicitud de trámites docentes se implementó siguiendo los principios del patrón arquitectónico de Drupal de n-capas (Ver Figura 1) (Drupal, 2023) descritas en el epígrafe 1.5 referente a las tecnologías. A continuación, se presenta esta arquitectura basada en capas teniendo en cuenta la arquitectura de la solución propuesta:

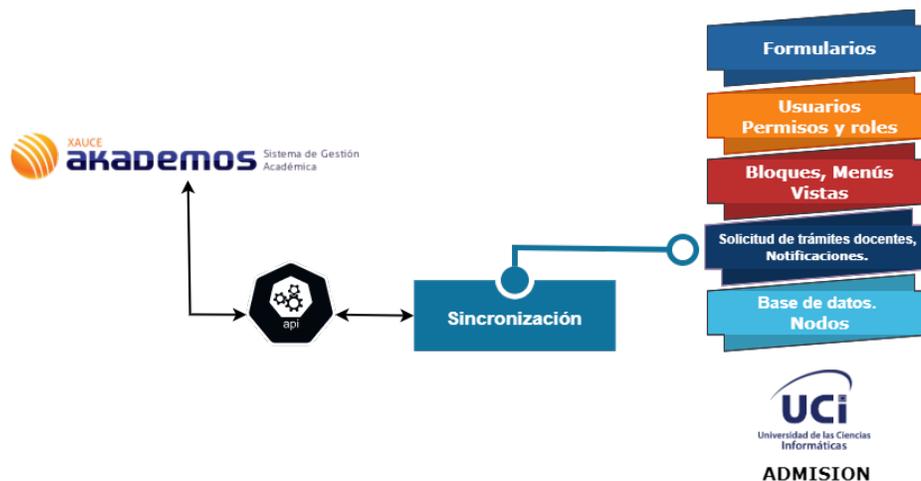


Figura 3. Representación arquitectónica. Fuente: elaboración propia.

Como se muestra en la figura 3 el módulo para solicitud de trámites docentes, así como sus respectivas configuraciones de notificaciones y sincronización entre bases de datos se encuentran ubicados dentro de la capa módulos de la arquitectura de Drupal 9. El módulo de sincronización hace uso de la API que es la herramienta para poder sincronizar los datos desde el Sistema de Admisión a la base de datos de Akademos. Se hace uso de la capa de datos, específicamente de su base de datos para realizar consultas al gestor utilizado. Estas capas representan el acceso a datos donde los usuarios externos no tienen permisos de acceso. En la capa de bloques y menús se crean los bloques, menús y vistas que serán renderizadas por las plantillas. Además, su interacción con la capa de permisos *de usuarios* donde se crean los permisos de acceso basado en roles. Por último, se tiene la capa de plantillas donde se realiza el manejo de formularios, y donde interactúa el cliente.

Para obtener una correcta implementación se hace necesario tener en cuenta las mejores soluciones conocidas, con el objetivo de diseñar las clases y las funcionalidades a codificar. Para ello se analizan y proponen algunos patrones de diseño que responden ante las necesidades de desarrollo.

2.4.2 Patrones de diseño

Los patrones de diseño, tratan los problemas que se repiten y que se presentan en situaciones particulares del diseño, con el fin de proponer soluciones a ellas. Se encargan de identificar clases, instancias, roles, colaboraciones entre estas, así como la distribución de responsabilidades (Larman, 2005). En resumen, es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular.

Los patrones de diseño orientado a objeto para la asignación de responsabilidades o comúnmente denominados patrones GRASP (de inglés *General Responsibility Assignment Software Patterns*), que como su nombre lo indica son los encargados de realizar responsabilidades según la función de cada uno de estos. Los patrones de diseño GRASP que se utilizan en el desarrollo del módulo son controlador, creador, alta cohesión y bajo acoplamiento.

- Controlador: se refiere a la clase responsable de manejar las interacciones del usuario y coordinar la lógica entre la vista y el modelo.

Como se muestra en la figura 4 el método `submitForm` procesa los valores enviados del formulario y actualiza la configuración correspondiente en el sistema. En este caso, actúa como un controlador para el formulario de configuración. Además, el método `buildForm` genera el formulario en sí, definiendo campos de configuración y sus valores por defecto.

```

/**
 * {@inheritdoc}
 * getFormId() - returns the form's unique ID
 */
public function getFormId() {
    return 'tramites_sincronizacion_form';
}

/**
 * {@inheritdoc}
 * buildForm() - returns the form array
 */
public function buildForm(array $form, FormStateInterface $form_state) {
    $config = $this->config('tramites_sincronizacion.adminsettings');
}

```

Figura 4. Patrón controlador. Fuente: elaboración propia.

- Creador: se refiere a la responsabilidad de un objeto o método para crear instancias de otro objeto.

Como se muestra en la figura 5 el método `nombreCompleto($user)` es responsable de construir y devolver un nombre completo a partir de los datos del usuario.

```

function nombreCompleto($user)
{
    $nombre = "";
    $nombre = $user->get('field_primer_nombre')->value;

    if (isset($user->get('field_segundo_nombre')->value) && !empty($user->get('field_segundo_nombre')->value))
        $nombre .= ' ' . $user->get('field_segundo_nombre')->value;
    }
    if (isset($user->get('field_primer_apellido')->value) && !empty($user->get('field_primer_apellido')->value))
        $nombre .= ' ' . $user->get('field_primer_apellido')->value;
    }
    if (isset($user->get('field_segundo_apellido')->value) && !empty($user->get('field_segundo_apellido')->value))
        $nombre .= ' ' . $user->get('field_segundo_apellido')->value;
    }
    return $nombre;
}

```

Figura 5. Patrón creador. Fuente: elaboración propia.

- Alta cohesión: implica que un conjunto de operaciones o responsabilidades de un módulo están estrechamente relacionadas, como se muestra en la figura 6.
- Bajo acoplamiento: indica que las piezas de código están menos interconectadas y dependen mínimamente entre sí.

```

function tramites_notificaciones_correo_solicitud_secretaria(&$form, FormStateInterface $form_state)
{
    //obtengo las configuraciones del modulo
    $config = \Drupal::config('tramites_notificaciones.adminsettings');

    //Obtengo los datos del usuario que creo la solicitud
    $user = User::load($form_state->getValue('uid')[0]['target_id']);
    $nombre = nombreCompleto($user);

    //Obtengo el correo de secretaria
    $correo = $config->get('notificaciones_tramites_correos_solicitud');

    //Obteniendo el tipo de trámite
    $node = $form_state->getFormObject()->getEntity();
    $tramite = $node->type->entity->label();
}

```

Figura 6. Patrón alta cohesión. Fuente: elaboración propia.

Los principales patrones de diseño son los conocidos como GOF (del inglés Gang Of Four, en español grupo de los cuatro).

A continuación, se hace referencia a los patrones que usa el CMS Drupal y los que serán usados en la implementación del módulo a desarrollar.

- Observador: los módulos que implementan un *hook* determinado por evento de inserción o actualización de una determinada entidad, son declarados como observadores de dichas entidades con las que interactúan.

Como se muestra en la figura 7 en el módulo se implementa el *hook*¹ *hook_form_alter ()* que se ejecuta cuando se realiza un cambio en los formularios.

¹ Son funciones PHP que aumentan la funcionalidad de Drupal permitiendo a los módulos interactuar con el Core. De esta manera se puede alterar los comportamientos default de Drupal, y agregar nueva funcionalidad, si esos son los requerimientos para el sitio.

```

function tramites_notificaciones_form_alter(&$form, FormStateInterface $form_state, $form_id)
{
    //acciones cuando se crea un tramite
    if ($form_id == "node_certificacion_de_notas_form" || $form_id == "node_certifico_de_estudios t
    //envio de correo al solicitante
    $form['actions']['submit']['#submit'][] = 'tramites_notificaciones_correo_solicitud_solicitar
    $form['actions']['submit']['#submit'][] = 'tramites_notificaciones_correo_solicitud_secretari
    }

    //acciones cuando se edita un tramite existente
    if ($form_id == "node_certificacion_de_notas_edit_form" || $form_id == "node_certifico_de_estud
    $form['actions']['submit']['#submit'][] = 'tramites_notificaciones_correo_solicitud_estados';
    }
}

```

Figura 7. Patrón observador. Fuente: elaboración propia.

- Fábrica: define una interfaz para crear un objeto, pero deja que las subclasses decidan qué clase instanciar. De esta manera, delega la lógica de creación a las subclasses.

Como se muestra en la figura 8 en la función `getEditableConfigNames ()`, la clase `ConfigurarTramitesNotificacionesForm` define un método que devuelve un array con el nombre de la configuración editable. Este método actúa como un *Factory Method*, permitiendo que las subclasses decidan los nombres de configuración editables específicos.

```

class ConfigurarTramitesSincronizacionForm extends ConfigFormBase {
    /**
     * {@inheritdoc}
     * getEditableConfigNames() - gets the configuration name
     */
    protected function getEditableConfigNames() {
        return [
            'tramites_sincronizacion.adminsettings',
        ];
    }
}

```

Figura 8. Patrón fábrica. Fuente: elaboración propia.

- Comando: acciones que se desencadenan en respuesta a un evento o solicitud específica, para encapsular estas acciones como objetos, permitiendo su parametrización y ejecución posterior.

Como se muestra en la figura 9 el método `submitForm ()` actúa como un comando que procesa y aplica los valores del formulario a la configuración.

```

/**
 * {@inheritdoc}
 * submitForm() - processes the form submission
 */
public function submitForm(array &$form, FormStateInterface $form_state) {
    parent::submitForm($form, $form_state);

    $values = $form_state->getValues();

    $this->config('tramites_sincronizacion.adminsettings')
        ->set('tramites_sincronizacion_api_insertar', $values['tramites_sincronizacion_api_inse
        ->set('tramites_sincronizacion_api_listar', $values['tramites_sincronizacion_api_listar
        ->save();
    }

```

Figura 9. Patrón comando. Fuente: elaboración propia.

- Estrategia: permite definir una familia de algoritmos, encapsular cada uno de ellos y hacerlos intercambiables. Esto permite que el algoritmo varíe independientemente de los clientes que lo utilicen, como se muestra en la figura 10.

```

function tramites_notificaciones_correo_solicitud_secretaria(&$form,
FormStateInterface $form_state)
{
    //obtengo las configuraciones del modulo
    $config = \Drupal::config('tramites_notificaciones.adminsettings');

    //Obtengo los datos del usuario que creo la solicitud
    $user = User::load($form_state->getValue('uid')[0]['target_id']);
    $nombre = nombreCompleto($user);

    //Obtengo el correo de secretaria
    $correo = $config->get('notificaciones_tramites_correos_solicitud');

```

Figura 10. Patrón estrategia. Fuente: elaboración propia.

2.4.3 Tarjetas CRC

Las tarjetas CRC son parte de la metodología XP para el diseño de software orientado a objetos. La forma de diseño y organización es diseñar una tarjeta CRC (Clase-Responsabilidad-Colaboración) por cada historia de usuario, ya que brinda una funcionalidad directa al negocio. Una clase es una persona, evento, concepto, pantalla o reporte, las responsabilidades de una clase es la información que se conoce y las actividades que realiza por los métodos y atributos, mientras que los colaboradores de una clase son las demás clases

con las que trabaja en conjunto para llevar a cabo sus responsabilidades (Paucar Quile, 2019). A continuación, se muestra la primera tarjeta de cada iteración, las demás se contemplan en los anexos:

Tabla 14. Tarjeta CRC 1. Fuente: elaboración propia.

Tarjeta CRC	
Clase	tramites_validacion_de_fotocopias_de_titulo
Responsabilidades	Colaboradores
crear_tramites validación_de_fotocopias_de_titulo () editar_tramites validación_de_fotocopias_de_titulo () eliminar_tramites validación_de_fotocopias_de_titulo () visualizar_tramites validación_de_fotocopias_de_titulo ()	notificacion_correo tramites_sincronizacion

Tabla 15. Tarjeta CRC 5. Fuente: elaboración propia.

Tarjeta CRC	
Clase	notificacion_correo
Responsabilidades	Colaboradores
notificacion_correo (): envía correos automáticos a los solicitantes y la secretaria.	tramites_validacion_de_fotocopias_de_titulo tramites_certifico_de_estudios_terminados tramites_certificacion_de_notas tramites_subsanacion_de_errores

Tabla 16. Tarjeta CRC 6. Fuente: elaboración propia.

Tarjeta CRC	
Clase	tramites_sincronizacion
Responsabilidades	Colaboradores
tramites_sincronizacion (): permite la inserción de los datos desde el Sistema de Admisión mediante una API a la base de datos de AKADEMOS.	tramites_validacion_de_fotocopias_de_titulo tramites_certifico_de_estudios_terminados tramites_certificacion_de_notas tramites_subsanacion_de_errores

2.4.4 Modelo de despliegue

Un diagrama de despliegue es un tipo de diagrama del lenguaje unificado de modelado que se utiliza para modelar el hardware utilizado en el despliegue del sistema y las relaciones entre

sus componentes. Muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos (Pressman, 2010).

En el diagrama que se muestra en la figura 11 se puede observar que el sistema cuenta con 5 nodos principales. El nodo PC cliente que requiere de un navegador web y se conecta al servidor web por el protocolo https:443/http:80. El nodo servidor web donde debe estar instalado el CMS Drupal 9, se conecta al servidor de bases de datos por el puerto 5432 y al servidor de correo SMTP: 587. Además, está conectado al nodo servidor de base de datos (API) por el puerto 8000. El nodo PC cliente representa las estaciones de trabajo de los usuarios que se conectan al CMS.

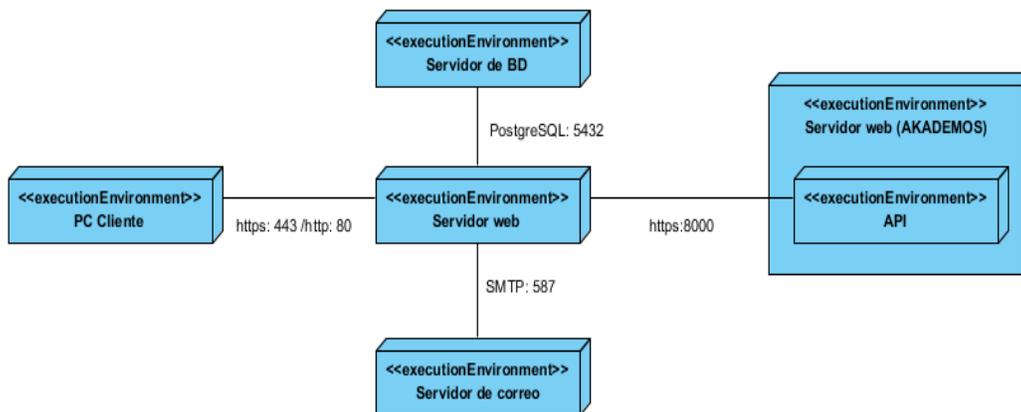


Figura 11. Diagrama de despliegue. Fuente: elaboración propia.

Conclusiones del capítulo

En este capítulo se han abordado los aspectos referentes a la concepción del producto a desarrollar y sus características funcionales. Esto permitió arribar a las siguientes conclusiones:

- A partir de la definición de las HU se pudieron identificar las principales funcionalidades a desarrollar en correspondencia al módulo de solicitudes de trámites docentes para el Sistema de Admisión.
- La estimación del tiempo para la implementación de las HU definidas, permitió calcular una entrega final del producto en doce semanas.

- Se identificaron los patrones estructurales y de comportamiento para el desarrollo de la solución, así como,
- Las tarjetas CRC permitieron determinar de las funciones, clases, instancias, roles y responsabilidades que maneja el módulo.
- El diagrama de despliegue permitió mostrar la arquitectura en tiempo de ejecución del sistema.

Con los aspectos arquitectónicos y de diseño a utilizar definidos, queda establecida la vía para llevar a cabo la implementación y prueba del módulo. Además, se adoptan los estándares de programación definidos por el equipo de desarrollo del Sistema de Admisión.

Capítulo 3. Implementación y prueba del módulo para solicitudes de trámites docentes.

Introducción

En el presente capítulo se detallan las iteraciones realizadas durante la etapa de construcción del módulo propuesto, además se exponen las tareas de ingeniería generadas para cada HU que fueron definidas, así como las pruebas de aceptación planificadas para el sistema. De esta forma es obtenido un producto funcional probado y listo para entregar al cliente al final de cada iteración como propone XP.

3.1 Fase III: Desarrollo.

En esta fase, XP plantea que las HU seleccionadas para ser implementadas se realizan durante el transcurso de la iteración a la que pertenecen. Por estas razones, se lleva a cabo una revisión del plan de iteraciones y se modifican en caso de ser necesario. Como parte de este plan se descomponen las HU en tareas de ingeniería (Joskowicz, 2015).

3.1.2 Tareas de ingeniería.

Las tareas de ingeniería pueden estar descritas por un lenguaje técnico y no ser necesariamente entendibles por el cliente. Tienen como objetivo definir cada una de las actividades que dan cumplimiento a las HU, de forma tal que se entienda lo que el sistema tiene que hacer y facilite su construcción. A continuación, se describen las tareas de ingeniería correspondientes a las HU del sistema. Para una mayor organización, se definen en correspondencia con las iteraciones definidas como se manifiesta a continuación.

Tareas de ingeniería para la Iteración I.

Para la primera iteración se definieron un total de 4 tareas de ingeniería, correspondientes a las HU de la 1 a la 16. Cada una de estas historias representa un trámite diferente, pero su funcionamiento es el mismo. Por ello se agruparon de la siguiente forma:

- Visualizar información de trámites docentes. (Validación de fotocopias del título, Certifico de estudios terminados, Certificación de notas y Subsanación de errores.)
- Crear solicitud de trámites docentes. (Validación de fotocopias del título, Certifico de estudios terminados, Certificación de notas y Subsanación de errores.)

- Editar solicitud de trámites docentes. (Validación de fotocopias del título, Certificado de estudios terminados, Certificación de notas y Subsanación de errores.)
- Eliminar solicitud de trámites docentes. (Validación de fotocopias del título, Certificado de estudios terminados, Certificación de notas y Subsanación de errores.)

A continuación, se describen las tareas realizadas en dicha iteración.

Tabla 17. Tarea de ingeniería 1. Iteración 1. Fuente: elaboración propia.

Tarea de Ingeniería	
Número de tarea: 1	Número de HU: 1, 5, 9, 13
Nombre de la tarea: Desarrollar la funcionalidad para visualizar la información de las solicitudes de trámites docentes.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 11/09/23	Fecha de fin: 22/09/23
Programador responsable: Lisette Oliva Alvarez	
Descripción: Para llevar a cabo la funcionalidad de visualizar la información de solicitudes de trámites docentes, primeramente, se creó la vista para mostrar la información de cada trámite. Ubicada en la carpeta admision-master\modules\custom\tramites_docentes. Una vez creadas todas las vistas, se procede a crear todas las configuraciones desde la carpeta \config\install para cada uno de los trámites.	

Tabla 18. Tarea de ingeniería 2. Iteración 1. Fuente: elaboración propia.

Tarea de Ingeniería	
Número de tarea: 2	Número de HU: 2, 6, 10, 14
Nombre de la tarea: Desarrollar funcionalidad para crear solicitudes de trámites docentes	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 25/09/23	Fecha de fin: 06/10/23
Programador responsable: Rolando Falcon Ochoa.	
Descripción: Para llevar a cabo la funcionalidad de crear solicitudes de trámites docentes primeramente se realizaron las validaciones pertinentes para que solo se mostrará el botón "solicitar trámite" a los egresados UCI. Posteriormente se crea el formulario correspondiente en la carpeta \config\install.	

Tabla 19. Tarea ingeniería 3. Iteración 1. Fuente: elaboración propia.

Tarea de Ingeniería	
Número de tarea: 3	Número de HU: 3, 7, 11, 15
Nombre de la tarea: Desarrollar funcionalidad para editar solicitudes de trámites docentes.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 09/10/23	Fecha de fin: 20/10/23
Programador responsable: Lisette Oliva Alvarez	
<p>Descripción: Para llevar a cabo la funcionalidad de editar solicitudes de trámites docentes primeramente se realizaron las validaciones pertinentes para que solo se mostrará el botón "solicitar trámite" a los egresados UCI. Posteriormente se crea el formulario correspondiente en la carpeta \config\install.</p>	

Tabla 20. Tarea de ingeniería 4. Iteración 1. Fuente: elaboración propia.

Tarea de Ingeniería	
Número de tarea: 4	Número de HU: 4, 8, 12, 16
Nombre de la tarea: Desarrollar funcionalidad para eliminar solicitudes de trámites docentes.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 23/10/23	Fecha de fin: 3/11/23
Programador responsable: Lisette Oliva Alvarez	
<p>Descripción: Para llevar a cabo la funcionalidad de eliminar solicitudes de trámites docentes primeramente se realizaron las validaciones pertinentes para que solo se mostrará el botón "solicitar trámite" a los egresados UCI. Posteriormente se crea el formulario correspondiente en la carpeta \config\install.</p>	

Tareas de ingeniería para la Iteración II

Para una segunda iteración, se elaboraron un total de 2 tareas de ingeniería. Una que contendrá las HU 17 y 18, y otra para la HU 19. A continuación, se describen las tareas realizadas en dicha iteración.

Tabla 21. Tarea de ingeniería 5. Iteración 2. Fuente: elaboración propia.

Tarea de Ingeniería	
Número de tarea: 5	Número de HU: 17, 18

Nombre de la tarea: Implementar funcionalidad para enviar notificaciones por correo electrónico a los solicitantes y a la secretaria general.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 6/11/23	Fecha de fin: 17/11/23
Programador responsable: Rolando Falcón Ochoa	
<p>Descripción: Para implementar la funcionalidad que permite notificar mediante el correo a los solicitantes y a la secretaria general cada vez que se solicite un trámite desde el Sistema de Admisión es necesario crear el módulo tramites_notificaciones ubicado en la carpeta \admission-master\modules\custom\tramites_docentes\tramites_notificaciones. Luego se crea la vista solo disponible para administradores donde se configuran las notificaciones ubicada en la carpeta \tramites_notificaciones\src\Form\ConfigurarTramitesNotificacionesForm.php. Se crean las funciones y validaciones necesarias para el correcto funcionamiento del módulo.</p>	

Tabla 22. Tarea de ingeniería 6. Iteración 2. Fuente: elaboración propia.

Tarea de Ingeniería	
Número de tarea: 6	Número de HU: 19
Nombre de la tarea: Desarrollar la funcionalidad para visualizar solicitudes de trámites docentes.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 20/11/23	Fecha de fin: 24/11/23
Programador responsable: Lisette Oliva Alvarez	
<p>Descripción: Para llevar a cabo la funcionalidad de visualizar solicitudes de trámites docentes, primeramente, se creó la vista para mostrar el listado de trámites. Ubicada en la carpeta \admission-master\modules\custom\tramites_docentes\lista_de_tramites. Una vez creada toda la vista, se procede a crear todas las configuraciones desde la carpeta \config\install\ en el archivo \views.view.tramites.yml donde se crea el bloque correspondiente a dicha información.</p>	

Con las tareas de ingeniería definidas, se hace necesario para el proceso de desarrollo de software, explicar los estándares de codificación seguidos. Esto es importante debido a que los estándares de codificación establecen reglas y pautas que los desarrolladores deben seguir al escribir código.

3.1.3 Estándares de codificación

La adopción de estándares de codificación constituye un faro conductor en la ruta hacia un código limpio, coherente y fácilmente mantenible. Estos estándares son directrices

predefinidas que estructuran y delimitan la escritura del código en un proyecto (Buytaert, 2005). Su valor radica en la coherencia y la claridad que infunden al código fuente.

La aplicación de estándares en los frameworks y CMS de desarrollo resulta fundamental para asegurar la calidad, cohesión y eficiencia en el desarrollo de software. Facilita a los desarrolladores el mantenimiento y la escalabilidad exitosa de proyectos, y asegura el cumplimiento de requisitos y regulaciones específicas de la industria (Drupal Association, 2018).

Drupal, reconocido por su robustez y comunidad activa, promueve la adopción de estándares de codificación para mantener la coherencia y calidad en su código fuente. La plataforma tiene su propia guía de estilo y convenciones que se deben seguir para garantizar la uniformidad en los proyectos basados en ella. A continuación, se presentarán ejemplos concretos de los estándares de codificación aplicados en el desarrollo del módulo para solicitudes de trámites docentes en Drupal 9, desde la nomenclatura de rutas, la organización de los nombres y estructura de las clases hasta el manejo de errores.

Identación

Para que el código sea más legible se puede insertar espacios en blanco. En Drupal se usa indentar con 2 espacios, y nunca con tabulaciones, también se debe tener presente que nunca se debe dejar espacios en blanco al final de cada línea, si se desea continuar en la siguiente línea simplemente se presiona la tecla `<ENTER>` (Drupal Association, 2018).

Estructuras de control

Las estructuras de control deben tener:

- Espacio entre las palabras reservadas (*if*, *while*, *for*, *else*, entre otros) y el paréntesis de apertura.
- Entre el paréntesis de cierre y la llave de apertura {también tiene su correspondiente espacio.
- Siempre se debe usar las llaves de apertura y cierre.
- Las estructuras *else* y *elseif* deben ir en líneas aparte, debajo del cierre de la estructura, en este caso debajo de la llave de cierre como se muestra en las figuras 11 y 12 respectivamente (Drupal Association, 2018).

```

// Crear el objeto que envía el correo
$mailManager = \Drupal::service('plugin.manager.mail');
$result = $mailManager->mail($module, $key, $to, $langcode, $params, NULL, $send);

if ($result['result'] !== TRUE) {
  \Drupal::messenger()->addError('Hubo un problema al enviar la notificación de correo al autor de esta solicitud.');
```

Figura 12. Estructuras de control. Fuente: elaboración propia.

```

// Obtener el mensaje predeterminado para el estado actual de la solicitud
$config = \Drupal::config('tramites_notificaciones.adminsettings');
switch ($estado) {
  case 'solicitado':
    $mensaje = $config->get('notificaciones_tramites_solicitado.value');
    break;
  case 'revisado':
    $mensaje = $config->get('notificaciones_tramites_revisado.value');
    break;
  case 'entregado':
    $mensaje = $config->get('notificaciones_tramites_entregado.value');
    break;
  default:
    $mensaje = '';
```

Figura 13. Estructura de control. Fuente: elaboración propia.

Etiquetas de apertura y cierre de PHP

Al escribir código PHP, se debe utilizar las etiquetas de apertura, en los archivos *.module* y *.inc* se puede omitir la etiqueta de cierre siempre y cuando se trabaje solo con PHP, omitir la etiqueta de cierre evita que queden líneas en blanco lo que podría dar como resultado un error típico "Cannot modify header information - headers already sent by ...". Cuando se trabaja con PHP puro, se puede evitar la etiqueta de cierre, pero dentro de HTML se debe utilizar la etiqueta de apertura de PHP (Drupal Association, 2018).

Comentar el código

Se debe diferenciar los comentarios que se utilizan para aclarar determinados bloques de código, ya que estos se pueden colocar donde se crea que sea necesario, de los comentarios de documentación que son los que se escriben al principio del archivo, o antes de declarar alguna función, estos últimos son sumamente importantes ya que por medio de una *API* se genera la documentación de ayuda desde las etiquetas que se emplean. En el tipo de

documentación que se usa para aclarar algún bloque de código se utiliza las etiquetas de apertura `/*` y de cierre `*/`, entre ellas se pueden colocar varias líneas de comentario, o se puede utilizar las etiquetas `//` para un comentario de una simple línea como se muestra en la figura 13 (Drupal Association, 2018)

```
/**
 * Implements hook_form_alter().
 */
0 references
function tramites_notificaciones_form_alter(&$form, FormStateInterface $form_state, $form_id)
{
    // echo $form_id;
    //acciones cuando se crea un tramite
}
```

Figura 14. Etiquetas y comentarios. Fuente: elaboración propia.

Uso de comillas

Como se muestra en la figura 14 se puede utilizar comillas simples ('texto') o comillas dobles ("texto") para delimitar las cadenas, las comillas dobles permiten interpretar el valor de una variable, entonces para "\$title", el resultado sería el valor de la variable \$title (Drupal Association, 2018).

Uso de punto y coma

Como se muestra en la figura 14 en Drupal siempre se debe finalizar las líneas de código PHP con el punto y coma (Drupal Association, 2018).

Operadores

Como se muestra en la figura 14 todos los operadores binarios (operadores que se utilizan entre dos valores), deben tener un espacio de separación entre ellos, esto aplica a operadores como:

`+`, `-`, `*`, `/`, `=`, `==`, `!=`, `,`, `.`, `+=`, `+-`, entre otros (Drupal Association, 2018).

Funciones

Como se muestra en la figura 14 los nombres de las funciones deben estar escritas en minúsculas, y las palabras separadas por un guion bajo. Siempre se debe escribir como prefijo el nombre del módulo, tema, entre otros para evitar así generar nombres duplicados. Entre el

nombre de la función y el paréntesis de inicio de los argumentos no se debe dejar espacios, cada argumento debe estar separados por una coma y luego un espacio. Se debe colocar un espacio entre el cierre del paréntesis y la apertura de la llave que contiene el bloque de código (Drupal Association, 2018).

```
function tramites_notificaciones_form_alter(&$form, FormStateInterface $form_state, $form_id)
{
    // echo $form_id;

    //acciones cuando se crea un tramite
    if ($form_id == "node_certificacion_de_notas_form" || $form_id == "node_certifico_de_estudios_terminad
    //envio de correo al solicitante
    $form['actions']['submit']['#submit'][] = 'tramites_notificaciones_correo_solicitud_solicitante';
    $form['actions']['submit']['#submit'][] = 'tramites_notificaciones_correo_solicitud_secretaria';
}

//acciones cuando se edita un tramite existente
if ($form_id == "node_certificacion_de_notas_edit_form" || $form_id == "node_certifico_de_estudios_ter
    $form['actions']['submit']['#submit'][] = 'tramites_notificaciones_correo_solicitud_estados';
}
}
```

Figura 15. Uso de comillas. Uso de punto y coma. Operadores. Funciones. Fuente: elaboración propia.

Nombre y descripción de los módulos

Por norma general, el nombre de un módulo nunca debe incluir guiones bajos, aunque este se componga de varias palabras. Pueden encontrarse nombres de módulos con guiones bajos ya que esta no es una regla obligatoria como se muestra en la figura 15. (Drupal Association, 2018).

```
1 name: Trámites - Notificaciones
2 type: module
3 description: Módulo para gestionar las notificaciones sobre las solicitudes de trámites.
```

Figura 16. Nombre y descripción de los módulos. Fuente: elaboración propia.

Definición de Rutas

Como se muestra en la figura 16 se utiliza el formato *yaml* para definir las rutas de administración del módulo (Drupal Association, 2018).

Uso de nombres de clases y métodos

Como se muestra en la figura 16 se hace referencia a la clase ``\Drupal\tramites_sincronizacion\Form\ConfigurarTramitesSincronizacionForm`` que corresponde al formulario de configuración del módulo (Drupal Association, 2018).

Requisitos de permisos

Como se muestra en la figura 16 se establecen requisitos de permisos para acceder a esta ruta específica en el administrador (Drupal Association, 2018).

Opciones de Ruta

Como se muestra en la figura 16 se establece `_admin_route: TRUE` para marcar esta ruta como una ruta de administración en Drupal (Drupal Association, 2018).

```
1  tramites_notificaciones.admin_settings_form:
2  path: '/admin/config/tramites_notificaciones/adminsettings'
3  defaults:
4  |   _form: '\Drupal\tramites_notificaciones\Form\ConfigurarTramitesNotificacionesForm'
5  |   _title: 'Configurar notificaciones de sobre los trámites'
6  requirements:
7  |   _permission: 'view the administration theme'
8  options:
9  |   _admin_route: TRUE
```

Figura 17. Rutas. Nombre de clases y métodos. Requisitos de permisos. Opciones de ruta.
Fuente: elaboración propia.

Nombre de los archivos

Como se muestra en la figura 17 siempre se debe escribir los nombres de archivos en minúsculas, salvo los nombres de archivos que contengan documentación como, por ejemplo, `README.txt`, `INSTALL.txt`, entre otros (Drupal Association, 2018).

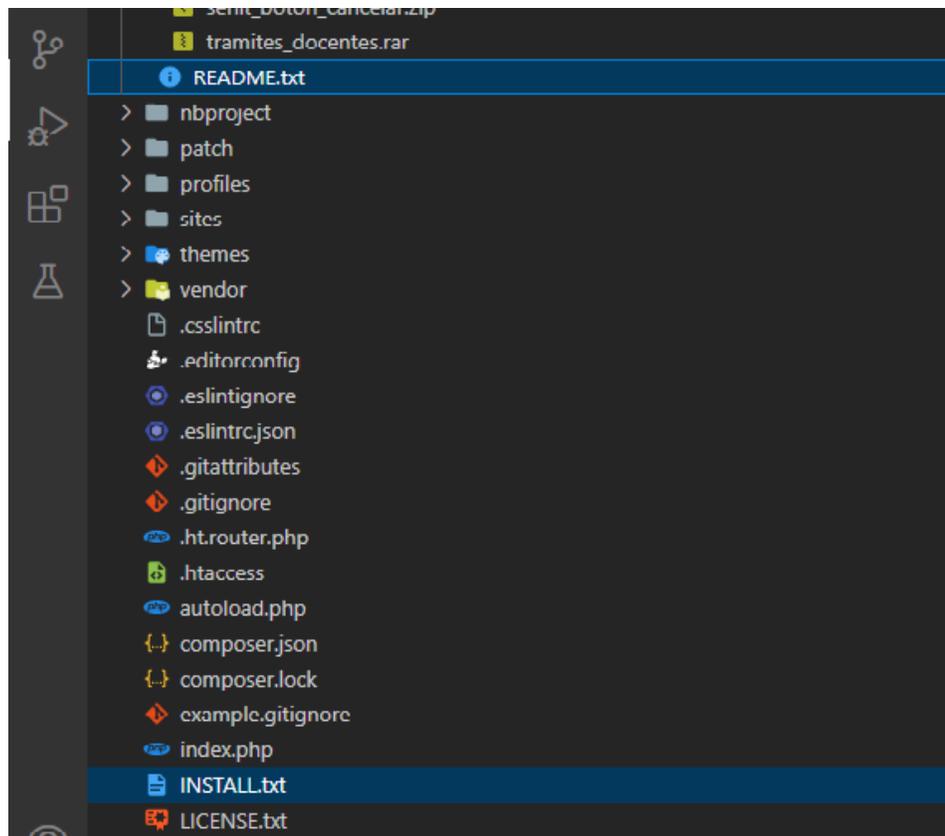


Figura 18. Nombre y descripción de los módulos. Fuente: elaboración propia.

Las aplicaciones de estos estándares no solo promueven la coherencia en el código, sino que también mejoran la legibilidad, facilitan el mantenimiento y reducen la probabilidad de errores. Luego de haber definido los estándares de codificación utilizados en el sistema, el siguiente paso es iniciar la fase de pruebas al sistema. Es esencial crear un conjunto de pruebas para evaluar la eficacia de la solución que se implementará. Lo que permitirá medir el nivel de cumplimiento con los objetivos de implementación trazados y el nivel de satisfacción del cliente.

3.2 Fase IV: Pruebas

Las pruebas, son una etapa del proceso de desarrollo del producto donde se asegura la calidad y el correcto funcionamiento del producto. Esta etapa tiene como objetivo verificar y validar que el sistema cumpla con sus especificaciones y funcionalidades definidas por el cliente (Diaz et al., 2020).

Las pruebas de software son una parte fundamental del ciclo de desarrollo de software. Se aplican en diferentes niveles y escenarios para alcanzar diferentes objetivos. Estos niveles o escenarios de pruebas se utilizan para asegurarse de que el software funcione de manera confiable y cumpla con los requisitos establecidos (Díaz et al., 2020). A continuación, se establecerán las pruebas que se les estarán realizando al módulo para solicitudes de trámites docentes del Sistema de Admisión de la UCI:

Tabla 23. Pruebas. Fuente: elaboración Tomado de (Díaz et al., 2020).

Pruebas	Objetivo	Etapa de ejecución
Pruebas funcionales	<ul style="list-style-type: none"> ● Validar y verificar que el producto cumple con lo especificado. ● Comprobar que el software hace lo que debe y cómo lo tiene q hacer. ● Proveer del grado de calidad del software. 	Desarrollo
Pruebas de Integración:	<ul style="list-style-type: none"> ● Reducir el riesgo. ● Verificar los comportamientos funcionales y no funcionales de las interfaces. ● Encontrar defectos en la interacción entre componentes ● Evitar que los defectos escapen a otros niveles de pruebas. ● Verificar la integración entre los sistemas. 	Desarrollo
Pruebas de Aceptación:	<ul style="list-style-type: none"> ● Validación de que el sistema está completo y funcionará según los previsto. 	Despliegue

	<ul style="list-style-type: none"> • Verificar los comportamientos funcionales y no funcionales del sistema. • Satisfacer requisitos de usuarios, operación y legales o reglamentarios. 	
--	---	--

3.2.1 Pruebas de funcionalidad

Las pruebas funcionales son consideradas como la parte más importante del desarrollo. Ellas están desarrolladas bajo la perspectiva del usuario, confirmando que el sistema hace lo que los usuarios esperan que haga. Un error funcional en la aplicación puede tener consecuencias catastróficas, desde la pérdida de credibilidad de los clientes, hasta grandes pérdidas económicas (Peño, 2015).

El objetivo de las pruebas de la funcionalidad se centra en:

- Verificar la función del sistema al fijar la tensión en la validación de las funciones, métodos, servicios.

Sus metas se centran en validar que la aplicación:

- Cumpla con los requisitos funcionales especificados en el diseño de la solución.
- Cumpla con los requisitos no funcionales especificados en el diseño de la solución.
- Cumpla con las restricciones de entrada y salida.
- Cumpla íntegramente con la estructura referencial especificada (Peño, 2015).

Entre sus principales ventajas se encuentran:

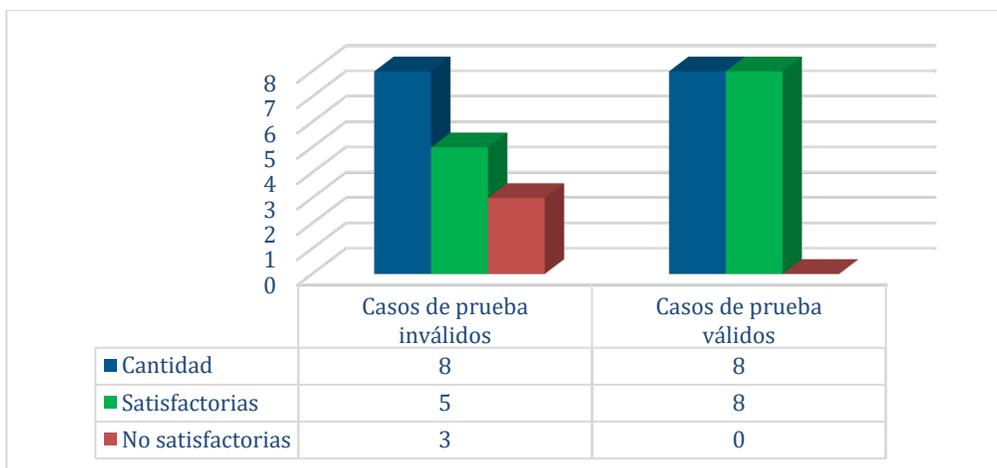
- Flexibilidad: Las pruebas funcionales no automatizadas permiten a los probadores adaptarse a los cambios en tiempo real durante el proceso de prueba.
- Pruebas exploratorias: Las pruebas funcionales no automatizadas permiten realizar pruebas exploratorias, que pueden ser útiles para encontrar defectos que no se detectarían con pruebas automatizadas.

- Pruebas de usabilidad: Las pruebas funcionales no automatizadas permiten a los probadores evaluar la usabilidad y la experiencia del usuario, lo cual puede ser difícil de lograr con pruebas unitarias automatizadas.
- Identificación de problemas de interfaz de usuario: Las pruebas funcionales no automatizadas pueden ser más efectivas para identificar problemas con la interfaz de usuario que pueden ser difíciles de detectar con pruebas unitarias automatizadas (Peño, 2015).

3.2.1.2 Análisis de las pruebas de funcionalidad

En el caso del módulo para solicitudes de trámites docentes desarrollado en Drupal 9 con las pruebas funcionales realizadas se pudo ir validando en tiempo real casos válidos e inválidos gracias a las prestaciones que brinda este CMS. Una de las más usadas fue *var_dump*, que es una función de PHP que permite imprimir en pantalla el contenido y la estructura de una variable, un objeto, un array, etc. Esta función es útil para depurar el código y ver los valores que se están manejando en cada momento sabiendo los valores esperados.

Para ejecutar estas pruebas se seleccionaron los casos definidos para las pruebas de aceptación descritos en el epígrafe 3.2.3 pues una vez validados por el equipo de desarrollo podían pasar a ser validados por el cliente. El resultado obtenido en esta etapa fue satisfactorio pues se mostraron los resultados esperados para cada caso. Por ejemplo, para la configuración del envío de correos como puede verse en la figura 25 que se encuentran en los anexos. Finalmente, se añadieron casos de prueba comprobando el submódulo de sincronización obteniendo un total de 16 casos de prueba (8 válidos y 8 inválidos) como puede observarse en la gráfica 1.



Gráfica 1. Análisis de pruebas funcionales. Fuente: elaboración propia.

Una vez finalizadas estas pruebas se está en condiciones de ejecutar las pruebas de integración al módulo.

3.2.2 Pruebas de integración

Las pruebas de integración representan una técnica fundamental que permite verificar la correcta interacción entre diferentes subsistemas y componentes de un proyecto. Estas pruebas aseguran que los distintos elementos funcionales se relacionan adecuadamente y cumplan con los requisitos establecidos, ya sean estos funcionales o no (Díaz et al., 2020).

El módulo de solicitudes de trámites docentes para el Sistema de Admisión de la UCI se integra con los entornos web desarrollados en Drupal 9 y se adapta al gestor de bases de datos PostgreSQL compatible con el CMS. Este módulo se despliega en el sistema operativo de Windows, en los distintos navegadores, potencializando su versatilidad y las configuraciones de Drupal. A continuación, se presenta en la tabla 24 como se realizaron las pruebas de integración específicas realizadas para asegurar la correcta interacción y funcionamiento del módulo.

Tabla 24. Prueba de integración con el CMS Drupal usando el gestor de base de datos PostgreSQL. Fuente: elaboración propia.

Caso de Prueba:	Prueba de integración con el CMS Drupal usando el gestor de base de datos PostgreSQL.
------------------------	---

Sistema/componente al que se integra:	CMS Drupal
Condiciones de ejecución:	Se debe tener instalado el gestor de datos de base de datos PostgreSQL y debe existir conexión con la misma. Además, se debe tener el entorno de desarrollo local XAMPP ejecutando el servidor Apache.
Descripción de la prueba:	Comprobar que el módulo cree en la base de datos del sitio la tabla correspondiente del módulo (trámites docentes) y el módulo se ejecute correctamente.
Pasos de ejecución:	El módulo se instala accediendo a la opción extender del panel de administración, una vez ahí se selecciona el módulo que se desea instalar, se guardan los cambios y se verifica que se haya instalado en admisión/admin/modules.
Resultados esperados:	Se debe instalar el módulo en el CMS Drupal usando el gestor de base de datos PostgreSQL correctamente sin ningún error.
Resultado:	Satisfactorio.

3.2.2.1 Análisis de las pruebas de integración

En el caso de la solución desarrollada, para la verificación de una correcta interoperabilidad entre los módulos desarrollados, se llevaron a cabo las acciones siguientes:

Integrar el módulo de solicitud de trámites docentes al Sistema de Admisión de la UCI. En las figuras 19 y 20 se muestra el proceso de instalación, en la 21 se muestra el resultado de la integración del módulo y en la 22 se muestra como se muestra el módulo una vez integrado.

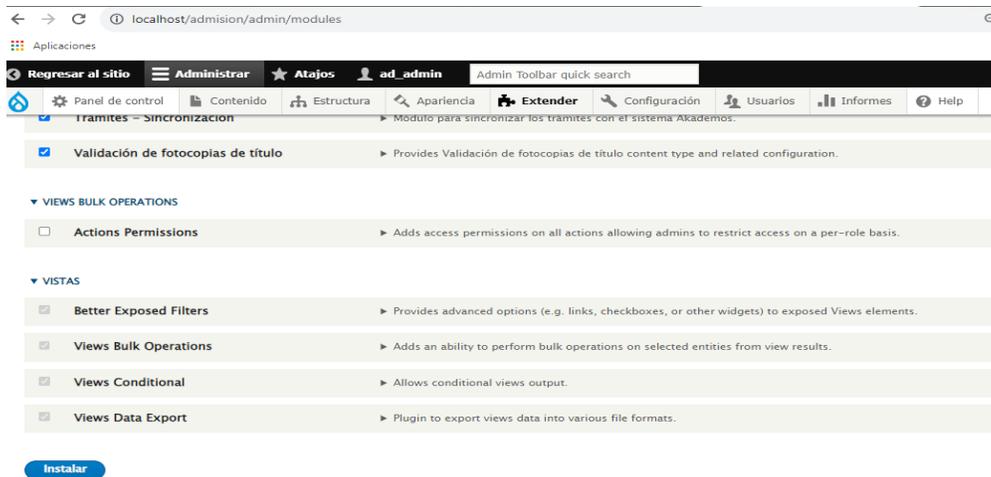


Figura 19. Prueba de integración. Fuente: elaboración propia.

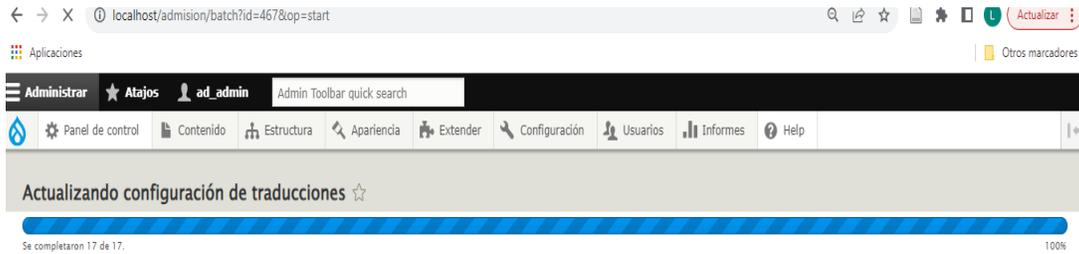


Figura 20. Prueba de integración. Fuente: elaboración propia.

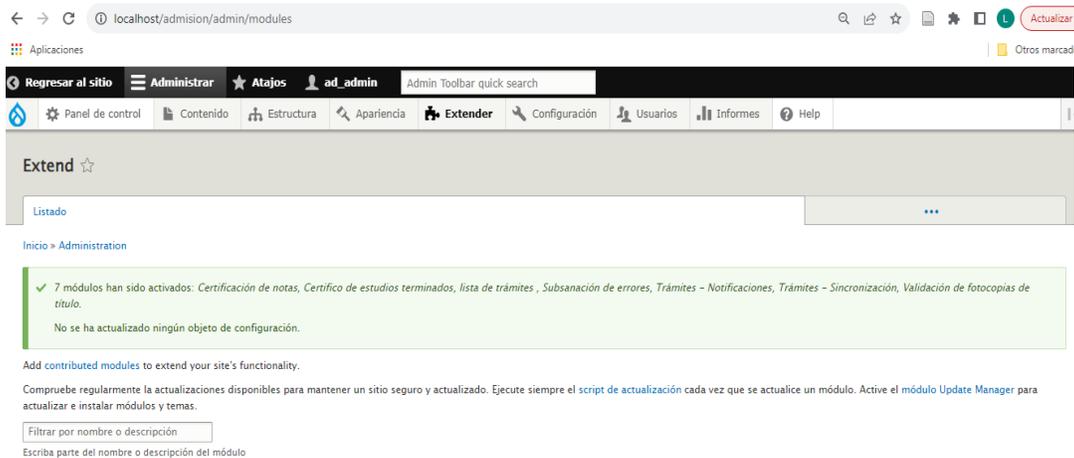


Figura 21. Resultado de las pruebas de integración. Fuente: elaboración propia.

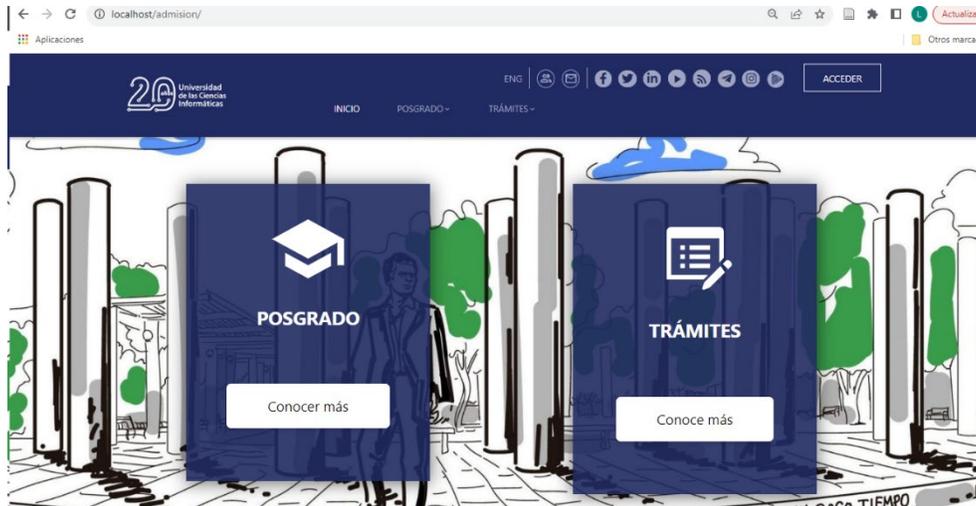


Figura 22. Módulo de Trámites. Fuente: elaboración propia.

3.2.3 Pruebas de aceptación

Las pruebas de aceptación se concentran en confirmar si el sistema completo cumple con los requisitos que se definieron por parte del cliente o las partes involucradas, ya sea en aspectos funcionales o no funcionales. Su principal objetivo es asegurar que el software sea útil y satisfaga las necesidades del usuario final. Estas pruebas se realizan como cajas negras, donde se esperan ciertas salidas del sistema. Es responsabilidad del cliente revisar y confirmar que estas pruebas sean correctas, tomando decisiones basadas en los resultados obtenidos (Crispin & Gregory, 2009).

Pruebas de aceptación para la Iteración I

Para la primera iteración, se definieron un total de 4 casos de pruebas de aceptación, pues se agruparon del mismo modo que para el desarrollo de las tareas de ingeniería. A continuación, se presenta la descripción de cada uno de ellos.

Tabla 25. Prueba de aceptación 1. Iteración 1. Fuente: elaboración propia.

Caso de prueba de aceptación	
Código: P1_HU1_5_9_13	Número de HU: 1, 5, 9, 13
Nombre: Visualizar información de validación de fotocopias de título.	
Descripción: Muestra la información necesaria a los usuarios sobre el proceso de solicitud y expedición de la validación de fotocopias del título.	

Condiciones de ejecución: Puede acceder cualquier usuario.
Pasos de ejecución: Para visualizar información de validación de fotocopias de título: <ol style="list-style-type: none"> 1. Acceder al Sistema de Admisión (https://admission.uci.cu/). 2. Todos los usuarios del sistema pueden acceder al módulo <i>Trámites</i>. 3. Todos los usuarios del sistema pueden visualizar la información de cada uno de los trámites.
Resultado esperado: Mostrar la información asociada y en caso de los usuarios registrados y autenticados UCI se muestra un botón para acceder al formulario de solicitud del trámite.
Evaluación: Satisfactorio

Tabla 26. Prueba de aceptación 2. Iteración 1. Fuente: elaboración propia.

Caso de prueba de aceptación	
Código: P2_HU_2_6_10_14	Número de HU: 2, 6, 10, 14
Nombre: Crear solicitud de trámites docentes.	
Descripción: Verifica que los usuarios con rol Usuario Autenticado y sean graduados uci tengan visible el botón “ <i>solicitar trámite</i> ” y al presionarlo se redirige hacia el formulario del trámite correspondiente.	
Condiciones de ejecución: Sólo pueden acceder los usuarios autenticados y que sean graduados UCI.	
Pasos de ejecución: Para crear solicitud de trámites docentes: <ol style="list-style-type: none"> 1. Acceder al Sistema de Admisión (https://admission.uci.cu/). 2. Iniciar sesión o registrarse en el sistema. 3. El usuario debe acceder al trámite deseado. 4. Hacer clic en el botón “<i>solicitar trámite</i>” y rellenar el formulario correspondiente. 5. Hacer clic en el botón “guardar”. 	
Resultado esperado: el módulo debe mostrar un mensaje notificando al usuario que su solicitud ha sido creada y que se ha enviado a su cuenta de correo.	
Evaluación: Satisfactorio	

Tabla 27. Prueba de aceptación 3. Iteración 1. Fuente: elaboración propia.

Caso de prueba de aceptación	
Código: P3_HU_3_7_11_15	Número de HU: 3, 7, 11, 15
Nombre: Editar solicitudes de trámites docentes	
Descripción: El administrador del sistema accede a la opción contenidos del panel de administración selecciona el trámite que desea editar y realiza los cambios en el formulario.	

Condiciones de ejecución: Solo puede acceder el administrador del sitio.
Pasos de ejecución: Para editar solicitudes de trámites docentes: <ol style="list-style-type: none"> 1. Acceder al Sistema de Admisión (https://admission.uci.cu/). 2. Registrarse como administrador 3. Ir a la opción contenido del panel de control. 4. Selecciona el trámite deseado. 5. Clic en el botón editar y modificar el formulario. 6. Guarda los cambios realizados.
Resultado esperado: el módulo debe permitir editar correctamente los trámites.
Evaluación: Satisfactorio

Tabla 28. Prueba de aceptación 4. Iteración 1. Fuente: elaboración propia.

Caso de prueba de aceptación	
Código: P4_HU_4_8_12_16	Número de HU: 4, 8, 12, 16
Nombre: Eliminar solicitudes de trámites docentes.	
Descripción: El administrador del sistema accede a la opción contenidos del panel de administración selecciona el trámite que desea eliminar y al presionar el botón eliminar se muestra un mensaje para verificar que si se desea eliminar ese trámite. Además de los botones eliminar y cancelar.	
Condiciones de ejecución: Sólo pueden acceder los usuarios autenticados que sean graduados UCI.	
Pasos de ejecución: Para visualizar información de validación de fotocopias de título: <ol style="list-style-type: none"> 1. Acceder al Sistema de Admisión (https://admission.uci.cu/). 2. Registrarse como administrador 3. Ir a la opción contenido del panel de control. 4. Selecciona el trámite deseado. 5. Clic en el botón eliminar. 	
Resultado esperado: el módulo debe permitir eliminar correctamente los trámites.	
Evaluación: Satisfactorio	

Pruebas de aceptación para la Iteración II

Para una segunda iteración, se diseñaron un total de 2 casos de pruebas de aceptación. Se agruparon las historias de usuario 17 y 18 y otro caso de prueba para la historia de usuario 19. Todas enfocadas a evaluar la implementación del módulo. A continuación, se describen las pruebas realizadas.

Tabla 29. Prueba de aceptación 5. Iteración 2. Fuente: elaboración propia.

Caso de prueba de aceptación	
Código: P17_HU_17_18	HU: 17,18
Nombre: Enviar notificaciones	
Descripción: el módulo debe enviar vía correo al usuario que se solicita y a la secretaria las notificaciones de que su solicitud fue creada correctamente y que desde el Sistema de Admisión se ha creado una nueva solicitud, respectivamente.	
Condiciones de ejecución: pueden recibir notificaciones todos los usuarios que realicen una solicitud y la secretaria cada vez que se cree una nueva solicitud.	
Pasos de ejecución: Acceder al Sistema de Admisión (https://admission.uci.cu/). <ol style="list-style-type: none"> 1. Iniciar sesión o registrarse en el sistema. 2. El usuario debe acceder al trámite deseado. 3. Hacer clic en el botón “solicitar trámite” y rellenar el formulario correspondiente. 4. Hacer clic en el botón “guardar”. 5. Enviar mensaje por correo notificando que su solicitud ha sido creada. 6. Enviar mensaje por correo notificando que se ha creado un nuevo trámite desde el Sistema de Admisión. 	
Resultado esperado: el módulo debe enviar notificaciones vía correo electrónico.	
Evaluación: Satisfactorio	

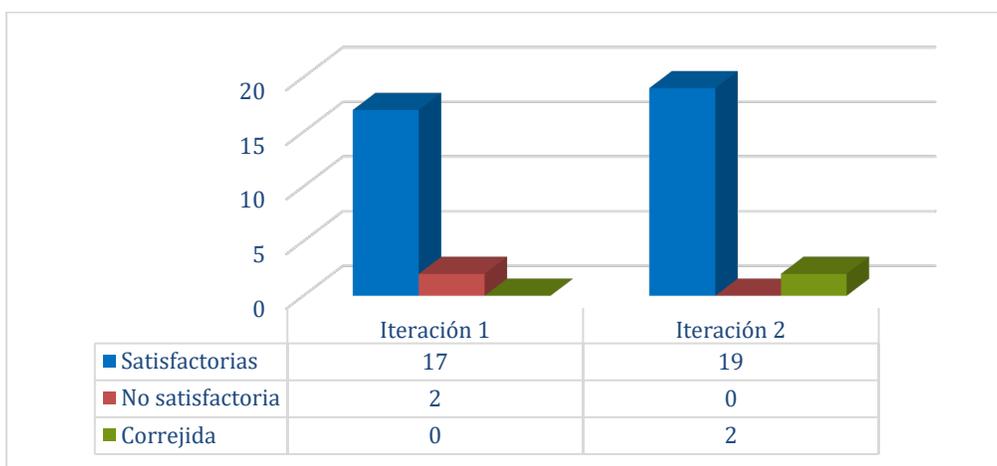
Tabla 30. Prueba de aceptación 6. Iteración 2. fuente: elaboración propia.

Caso de prueba de aceptación	
Código: P18_HU19	HU: 19
Nombre: Visualizar solicitudes de trámites docentes.	
Descripción: los usuarios autenticados y graduados UCI podrán ver en su perfil de usuario un listado de las solicitudes que haya realizado. Los datos que se mostrarán son: <ul style="list-style-type: none"> ● Tipo de solicitud. ● Motivo. ● Estado. ● Fecha de aprobación. ● Fecha de entrega. ● Validez. ● Fecha de solicitud. 	
Condiciones de ejecución: Sólo pueden acceder los usuarios autenticados que sean graduados uci que hayan realizado solicitudes.	

<p>Pasos de ejecución: Para visualizar solicitudes de trámites docentes:</p> <ol style="list-style-type: none"> 1. Acceder al Sistema de Admisión (https://admission.uci.cu/). 2. Iniciar sesión o registrarse en el sistema. 3. Crear una solicitud de trámite docente. 4. Entrar al “perfil de usuario” apartado “trámites”. 5. Visualizar la información de sus trámites.
<p>Resultado esperado: el módulo debe mostrar un listado de los trámites que ha creado el usuario.</p>
<p>Evaluación: Satisfactorio</p>

3.2.3.1 Análisis de las pruebas de aceptación

Se han desarrollado un total de 6 casos de pruebas de aceptación agrupando las historias de usuario según las prioridades y complejidad. Estas pruebas fueron realizadas de forma organizada, por cada iteración definida. A continuación, se muestra en un gráfico de barras, los porcentajes de satisfacción alcanzados en cada iteración.



Gráfica 2. Análisis de pruebas de aceptación. Fuente: elaboración propia.

Como se puede observar en la representación, en la primera iteración se realizaron un total de 6 casos de pruebas, de ellas alcanzaron el nivel de satisfacción esperado 17 requerimientos, mientras que, uno de los casos de pruebas donde se contemplan los requerimientos de envío de notificaciones, tuvo fallos que fueron detectados. Esto permitió corregirlos, e incorporar la seguridad de esta operación en el sistema. En tanto, en la iteración II, se probaron nuevamente los 6 casos de prueba respectivamente, donde todos los requerimientos fueron evaluados con un resultado satisfactorio. Con estas comprobaciones, se obtiene un 100 % de satisfacción en

la iteración final del producto, comprobando el correcto funcionamiento de las funcionalidades implementadas.

Conclusiones del capítulo

En este capítulo se han abordado los aspectos referentes a la implementación y validación del módulo. Esto permitió arribar a las siguientes conclusiones:

- El uso de los estándares de codificación para la implementación del módulo posibilita adoptar una estructura que contribuye a la calidad del módulo y obtener un código legible y fácil de comprender, con una menor cantidad de errores.
- Las pruebas al software permitieron detectar tempranamente las deficiencias del desarrollo, corregir errores y fallos en el funcionamiento del módulo, así como validar su correcto funcionamiento, para asegurar que la propuesta solución sea estable y segura.

Conclusiones

La presente investigación tuvo como objetivo principal el desarrollo de un módulo para la gestión de solicitudes de trámites docentes que se integrará al Sistema de Admisión de la UCI, utilizando la plataforma Drupal 9. Para ello, se realizaron las siguientes actividades:

- Estudio relacionado con los sistemas de gestión de solicitudes de trámites docentes, permitió establecer los criterios y conceptos fundamentales para el desarrollo del módulo.
- El análisis y diseño del módulo, utilizando la metodología XP, permitió obtener las historias de usuario como el artefacto necesario para el proceso de desarrollo.
- El módulo para la solicitud de trámites docentes, utilizando Drupal 9 como plataforma de desarrollo, permitió cumplir con los requerimientos funcionales identificados en la fase de planificación.
- Cumplió con los estándares de calidad y seguridad establecidos por la UCI.
- Se validó y verificó el correcto funcionamiento y la adecuación del módulo a las necesidades de los usuarios, mediante la ejecución de pruebas funcionales, de integración y de aceptación.
- Se realizó una *API* que permitió la sincronización entre el sistema de admisión y el sistema AKADEMOS.
- Facilitó el intercambio de información entre ambos sistemas y mejoró la eficiencia del proceso de solicitud de trámites docentes.
- Fue un valor añadido a la propuesta de solución, ya que no era un objetivo principal de la tesis, pero sí un aspecto relevante para la integración de los sistemas.

Como resultado de la investigación se obtuvo una solución eficiente para la gestión de solicitudes de trámites docentes en la UCI, que se integró al Sistema de Admisión existente y que mejoró la calidad del servicio y la satisfacción de los usuarios.

Recomendaciones

Una vez concluida la investigación y el desarrollo de la propuesta de solución, los autores del presente trabajo recomiendan:

- Añadir la funcionalidad de pago en línea para el servicio de impresión de documentos mediante el servicio de QR.

Referencias Bibliográficas

- Agile Lite. (2019). *Agile Lite: Ágil, sin todo el agotamiento*. <https://humanos.uci.cu/2019/07/03/agile-lite-sin-todo-el-agotamiento>
- Aubry, C. (2019). *HTML 5 y CSS 3: Revolucione el diseño de sus sitios web (4ta ed.)*. <http://desarrolloweb.com>
- Awad, M. A. (2005). *Comparison between Agile and Traditional Software Development Methodologie*.
- Balsamiq. (2023). *Balsamiq for Desktop—Fast, Powerful, Offline Wireframing | Balsamiq*. <https://balsamiq.com/wireframes/desktop/>
- Battaglia, N., Martínez, R., & Vincenz, M. C. D. (2017). *Una propuesta de evaluación de herramientas CASE para la enseñanza*.
- Beck, K., & Andres, C. (2012). *Extreme Programming Explained: Embrace Change*. (2nd ed.).
- Buytaert, D. (2005). *Creating and maintaining a high-performance Drupal site*. Drupal.org. <https://www.drupal.org/creating-maintaining>.
- Canós, J. H., Patricio Letelier, & Carmen Penadés. (2018). *Agile Methodologies in Software Development*. Camino de Vera s/n, 46022 Valencia. <http://www.aleteya.cs.buap.mx/~jlavalle/papers/agileMethodology/TodoAgil.pdf>
- Carbó, Y. M. (2016). *Cuba y el impacto de las TIC en la informatización de la sociedad cubana*. Monografía.com S.A. <http://monografias.com>
- Costa, G. I. (2007). Akademos, un Sistema Automatizado para la Gestión Académica. *Serie Científica de la Universidad de las Ciencias Informáticas*, 1(1), Article 1. <https://publicaciones.uci.cu/index.php/serie/article/view/253>
- Crispin, L., & Gregory, J. (2009). *Agile testing: A practical guide for testers and agile teams*. (1st ed.). Pearson Education, Inc. https://scrummalaysia.com/media/attachments/2020/03/18/agile_testing_-_a_practical_guide_for_testers_and_agile_teams.pdf
- Cruz-Amarán, D., & Guerrero, M. (2019). *La transformación digital y la universidad cubana* (pp. 394-402). https://www.researchgate.net/publication/339004202_La_transformacion_digital_y_la_universidad_cubana

- Decreto-Ley n° 370/2018, sobre la informatización de la Sociedad en Cuba—Informática Jurídica, (2019). <https://www.informatica-juridica.com/decreto-ley/decreto-ley-no-370-2018-sobre-la-informatizacion-de-la-sociedad-en-cuba/>
- Delgado, E. E. (2008). Metodologías de desarrollo de software. ¿Cuál es el camino? *Revista de Arquitectura e Ingeniería de Matanzas*, 2(3), 3.
- Díaz, A. M., Casañola, Y. T., & Denys Buedo Hidalgo. (2020). Estrategia de pruebas para organizaciones desarrolladoras de software. Testing strategy for software development organizations. *Revista Cubana de Ciencias Informáticas*, 14(3), 83-104.
- Dony, O. (2023, noviembre 9). *PostgreSQL: The World's Most Advanced Open Source Relational Database*. PostgreSQL. <https://www.postgresql.org/>
- Drupal. (2023a). *Overview of Drupal | Understanding Drupal | Drupal Wiki guide on Drupal.org*. <https://www.drupal.org/docs/understanding-drupal/overview-of-drupal>
- Drupal. (2023b). *PHPUnit in Drupal*. <https://www.drupal.org/docs/automated-testing/phpunit-in-drupal>
- Drupal. (2023c, octubre 26). *Coding standards*. Drupal.org. <https://www.drupal.org/docs/develop/standards>
- Drupal Association. (2018). *Drupal coding standards*. Drupal.org. <https://www.drupal.org/docs/develop/standards>.
- Enrique, G. S., & Milagro, O. V. (2019). *Las TIC como Herramientas de Comunicación entre Docentes y la Secretaría de*.
- Fernández, F. (2021). *Aplicaciones web con php*. Rama Editorial.
- GitLab. (2023, diciembre 6). *GitLab.org / GitLab · GitLab*. GitLab. <https://gitlab.com/gitlab-org/gitlab>
- González, V. M. V., Ramos, M. de la L. V., Reyes, D. del C. C., Contreras, D., & Baeza, T. de J. J. (2021, septiembre). Sistema gestor de contenidos ¿una herramienta más o un estándar? *Innovación y desarrollo tecnológico. Revista Digital*, 13(3), 13.
- Guzmán, M. A. (2020). *Taller: Implementación de sitios web con Drupal*. https://www.visibilidadweb.unam.mx/sites/default/files/docs/Taller_Drupal.pdf
- Hernandez, R. M. (2017). Impact of ICT on Education: Challenges and Perspectives. *Propósitos y Representaciones*, 5(1), Article 1. <https://doi.org/10.20511/pyr2017.v5n1.149>

- IONOS Digital Guide. (2023, junio 27). *Middleware: Explicación y aplicación*.
<https://www.ionos.es/digitalguide/servidores/know-how/middleware/>
- Joskowicz, J. (2015). *Reglas y prácticas en eXtreme Programming*. Universidad de Vigo. <http://jqueryhouse.com/best-jquery-notification-plugins/>
- Lamancha, B. P. (2004). Gestión de las Pruebas Funcionales. *Actas de Talleres de Ingeniería del Software y Bases de Datos*, 1(4).
https://d1wqtxts1xzle7.cloudfront.net/61308934/Gestion_de_Pruebas_Funcionales
- Larman, C. (2005). *Applying UML and patterns: An introduction to object-oriented analysis and iterative development*. Icaria Editorial.
- Lucid. (2023, junio 25). *What is Unified Modeling Language*. Lucidchart.
<https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>
- Marquez, M. C. G. (2022). *PHP. Definición de PHP, instalación de XAMPP, estructura de un programa en PHP, tipos de variables, operadores especiales, estructuras condicionales. Funciones. Arreglos. Procedimientos, PHP con CSS3. Formulario con PHP. Validación de formulario con PHP, requerimientos*. Universidad Nacional de Educación Enrique Guzmán y Valle.
- Mayo, R. B. Á., & Cantón, I. (2009, noviembre 10). Las tecnologías de la información y la comunicación en la educación superior. Estudio descriptivo y de revisión. *Revista Iberoamericana de Educación / Revista Ibero-americana de Educação*, 50/7.
- Meléndez, S. M. V., Gaitan, M. E., & Pérez, N. N. P. (2019). *Metodología ágil de desarrollo de software: Programación Extrema* [Universidad Nacional Autónoma de Nicaragua].
<https://repositorio.unan.edu.ni/1365/1/62161.pdf>
- Menzinsky, A., G. L. (2023). *Historias de Usuario. Ingeniería de requisitos ágil, 3.01 de Scrum*.
https://www.scrummanager.com/files/scrum_manager_historias_usuario.pdf
- Mescyt. (2022). Legalización de Documentos [Legalización de Documentos Académicos]. *Ministerio de Educación Superior, Ciencia y Tecnología*.
https://mescyt.gob.do/servicios__trashed/legalizacion-de-documentos/
- MINTEL. (2023, junio 7). *Las TIC como motor de la economía digital*. [Portal único de trámites ciudadanos]. Ministerio de Telecomunicaciones y de la Sociedad de la Información.
<https://www.telecomunicaciones.gob.ec/las-tic-como-motor-de-la-economia-digital/>

- Navarro, M. E., Moreno, M. P., Aranda, J., Parra, L., Rueda, J. R., & Pantano, J. C. (2017). *Integración de arquitectura de software en el ciclo de vida de las metodologías ágiles*. 566-596. <https://www.semanticscholar.org/paper/Integraci%C3%B3n-de-arquitectura-de-software-en-el-ciclo-Navarro-Moreno/c9de76eb31120f7d2abdfc0845edda0033504eb9>
- Ordóñez, M. P. Z., Ríos, J. R. M., & Castillo, F. F. R. (2017). *Administración de Bases de datos con PostgreSQL* (Vol. 19). 3Ciencias.
- Orozco, Righi, G., Ligia. (2018). *Integración Continua: Solución a los problemas de productividad y calidad del código en un entorno ágil testigo*. [Tesis de Grado]. Instituto Universitario Aeronáutico. <https://rdu.iua.edu.ar/bitstream/123456789/1781/1/TFG.pdf>
- Ortí, C. B. (2015). *Las Tecnologías de la Información y Comunicación (T.I.C.)*. <https://www.uv.es>
- Parada, C. J., Pilar Rojas Puentes, & Rodríguez, J. del P. (2019). Análisis de las Metodologías Ágiles para el Desarrollo de Software. *Investigación e Innovación en Ingeniería de Software*, 3, 9-24.
- Parra, A. (2018). *JavaScript*. <http://www.devolper.mozilla.org/>
- Paucar Quile, D. I. (2019). *Sistema informático para emprendimientos en la facultad de Ciencia e Ingeniería en Alimentos de la Universidad Técnica de Ambato y Comunidad*.
- Peño, J. M. S. (2015). *Pruebas de software. Fundamentos y técnicas*. (p. 130) [Tesis de Grado]. Universidad Politécnica de Madrid.
- Pérez, N. P. (2020). *Solución informática para la solución del servidor web durante la migración a código abierto*.
- PHP. (2023). *PHP: Documentation*. <https://www.php.net/docs.php>
- Powell, T. A. (2010). *HTML & CSS: The Complete Reference* (Fifth Edition). The McGraw-Hill Companies. <https://www.dcpehvpm.org/E-Content/BCA/BCA-II/Web%20Technology/the-complete-reference-html-css-fifth-edition.pdf>
- Pressman, R. S. (2010). *Ingeniería del software. Un enfoque práctico*. (Séptima). McGRAW-HILL.
- Puig, Y. (2021, diciembre 13). *De la informatización de la sociedad a la transformación digital en Cuba*. Presidencia de Cuba. <https://www.presidencia.gob.cu/es/noticias/de-la-informatizacion-de-la-sociedad-a-la-transformacion-digital-en-cuba/>

- Resolución No. 80/22, Pub. L. No. 80, 4 (2022).
- Robbio, A. P. (2019). *Los Joven Club cumplen 32 años de fundados*. [Diario Juventud Rebelde]. Fidel, Soldado de las ideas. <http://www.fidelcastro.cu>
- Sahoo, A. (2023). *pgAdmin is the most popular and feature rich Open Source administration and development platform for PostgreSQL, the most advanced Open Source database in the world*. pgAdmin PostgreSQL Tools. <https://www.pgadmin.org/docs/pgadmin4/development/>
- Sasidharan, A. (s. f.). *Patrones de diseño de software: Arquitectura y reutilización de Drupal* |. Recuperado 21 de septiembre de 2023, de <https://abh.ai/blog/software-design-patterns-drupals-architecture-and-reusability>
- Scrum. (2023, septiembre 14). *¿Qué es la metodología Scrum y cómo funciona?* <https://asana.com/es/resources/what-is-kanban>
- Sede Electrónica, U. (2023). *Inicio—Universidad de Castilla-La Mancha—Sede Electrónica*. <https://www.sede.uclm.es/>
- Segura, G. M. (2022, mayo 15). *Plataforma de la UNICA permite validar títulos en línea*. Periódico Invasor - Diario online de Ciego de Ávila. <http://www.invasor.cu/es/secciones/ciencia-y-tecnologia/plataforma-de-la-unica-permite-validar-titulos-en-linea>
- Shevts., A., Solomon, R., & Andrew Wetmore. (2019). *Dive into Design Patterns*. <https://www.goodreads.com/es/book/show/43125355>
- SIET. (2023). *UBA: Académicos*. Sistema integrado de expedición de títulos. <https://sietgraduados.rec.uba.ar/>
- Sistema de Admisión*. (2021). www.admision.uci.cu
- Sommerville, I. (2011). *Ingeniería de Software* (Novena). Pearson Educación.
- TAD-UBA. (2023). *Titulos—Facultad de Ciencias Sociales*. <https://www.sociales.uba.ar/titulos/>
- TIC Portal. (2022, septiembre 26). *Middleware: ¿cómo consigue que los sistemas hablen entre sí?* <https://www.ticportal.es/glosario-tic/middleware>
- UCI. (2021). *Universidad de las Ciencias Informáticas*. Sistema de Admisión. <http://www.admision.uci.cu>
- UCI. (2023a). *Universidad de las Ciencias Informáticas*. Portal, Universidad de las Ciencias Informáticas. <http://www.uci.cu>

- UCI. (2023b, julio 1). *XAUCE AKADEMOS 1.0* | *Universidad de las Ciencias Informáticas*.
<https://www.uci.cu/investigacion-y-desarrollo/productos/xauce/xauce-akademos-10>
- UCLM. (2023). *Sede Electrónica*. *Universidad de Castilla-la Mancha*.
<https://www.sede.uclm.es/>
- UNICA. (2022). *UniLegal*. <https://unilegal.unica.cu/>
- UNIR. (2023a, 02). Gestores de base de datos: Qué son y qué tipos existen. *Ingeniería*.
<https://mexico.unir.net/ingenieria/noticias/gestores-de-base-de-datos/>
- UNIR. (2023b, enero 9). ¿Qué es SASS y cómo aprenderlo? *Ingeniería y tecnología*.
<https://www.unir.net/ingenieria/revista/que-es-sass/>
- Visual Paradigm. (2021). *Visual Paradigm Online*. <http://www.visual-paradigm.com>
- VP Online. (2023, junio 25). *Ideal Modeling & Diagramming Tool for Agile Team Collaboration*.
<https://www.visual-paradigm.com/>
- XAMP. (2023, enero 3). *XAMPP tutorial: Create your own local test server*. Digital Guide
IONOS. <https://www.ionos.com/digitalguide/server/tools/xampp-tutorial-create-your-own-local-test-server/>
- XAMPP Hosting. (2022). *Apache HTTP Server Version 2.2 Documentation*.
<https://httpd.apache.org/docs/2.2/en/>

Anexos

Historias de Usuario

Tabla 31. Historia de Usuario 5. Fuente: elaboración propia.

Historia de Usuario	
Número: 5	Nombre: Visualizar información de certificado de estudios terminados.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos de estimación: 0.4	Iteración asignada: 1
<p>Descripción: El módulo debe permitir visualizar la información sobre los pasos y procedimientos para realizar la solicitud del certificado de estudios terminados.</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> ● Esta información es visible para todo tipo de usuario. ● El botón de “solicitar trámite” no está disponible para usuarios que no estén autenticados. ● El botón de “solicitar trámite” no está disponible para usuarios que no son graduados UCI. 	
<p>Interfaz:</p> 	

Tabla 32. Historia de usuario 6. Fuente: elaboración propia.

Historia de Usuario	
Número: 6	Nombre: Crear solicitud de certificado de estudios terminados.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos de estimación: 0.5	Iteración asignada: 1

Descripción: El usuario con rol usuario autenticado y graduado UCI al acceder al módulo debe visualizar el botón “solicitar trámite” al presionarlo aparecerá un formulario con los campos necesarios para crear el trámite. Debe poseer además los botones de guardar y cancelar respectivamente.

Los datos del formulario son:

- Tipo de solicitud.
- Motivo.
- Estado.
- Fecha de aprobación.
- Fecha de entrega.
- Validez
- Fecha de solicitud.

Observaciones:

- Al guardar el formulario se envía un correo al usuario confirmando la creación de su solicitud, y la secretaria notificando la nueva solicitud.

Interfaz:

The screenshot shows a web form with the following elements:

- Tipo de Solicitud:** A dropdown menu with options 'Entidad' and 'Personal'.
- Fecha de Solicitud:** A date input field showing '15/09/2023' with a calendar icon.
- Motivo:** A large empty text input box.
- Validez:** A dropdown menu with options 'Nacional' and 'Internacional'.
- Fecha de Aprobación:** A date input field showing '15/09/2023' with a calendar icon.
- Fecha de Aprobación:** A second date input field showing '15/09/2023' with a calendar icon.
- Buttons:** Two buttons labeled 'Cancelar' and 'Guardar'.
- Logo:** A logo consisting of three overlapping squares in the bottom left corner.

Tabla 33. Historia de usuario 7. Fuente elaboración propia.

Historia de Usuario	
Número: 7	Nombre: Editar solicitud de certificado de estudios terminados.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos de estimación: 0.3	Iteración asignada: 1
<p>Descripción: El usuario con rol de administrador accede a la opción de contenidos en el panel de administración al presionar el botón editar se debe mostrar el formulario para modificar los campos necesarios del trámite junto a los botones guardar y cancelar.</p> <p>Los datos del formulario son:</p> <ul style="list-style-type: none"> • Tipo de solicitud. • Motivo. • Validez. 	

Observaciones:

- El formulario está disponible solo para usuarios con rol administrador.

Tabla 34. Historia de usuario 8. Fuente: elaboración propia.

Historia de Usuario	
Número: 8	Nombre: Eliminar solicitud de certificado de estudios terminados.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de estimación: 0.3	Iteración asignada: 1
Descripción: El usuario con rol de administrador accede a la opción de contenidos en el panel de administración, selecciona el trámite que desea eliminar y al presionar el botón de eliminar se muestra un mensaje para verificar si ese es el que se quiere eliminar. Además, se muestran los botones de eliminar y cancelar respectivamente.	
Observaciones: <ul style="list-style-type: none"> • El formulario está disponible solo para usuarios con rol administrador. 	

Tabla 35. Historia de usuario 9. Fuente: elaboración propia.

Historia de Usuario	
Número: 9	Nombre: Visualizar información de certificación de notas.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos de estimación: 0.4	Iteración asignada: 1
Descripción: El módulo debe permitir visualizar la información sobre los pasos y procedimientos para realizar la solicitud de la certificación de notas.	
Observaciones: <ul style="list-style-type: none"> • Esta información es visible para todo tipo de usuario. • El botón de “solicitar trámite” no está disponible para usuarios que no estén autenticados. • El botón de “solicitar trámite” no está disponible para usuarios que no son graduados UCI. 	



Tabla 36. Historia de usuario 10. Fuente: elaboración propia.

Historia de Usuario	
Número: 10	Nombre: Crear solicitud de certificación de notas.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos de estimación: 0.5	Iteración asignada: 1
<p>Descripción: El usuario con rol usuario autenticado y graduado UCI al acceder al módulo debe visualizar el botón “solicitar trámite” al presionarlo aparecerá un formulario con los campos necesarios para crear el trámite. Debe poseer además los botones de guardar y cancelar respectivamente. Los datos del formulario son:</p> <ul style="list-style-type: none"> • Tipo de solicitud. • Motivo. • Estado. • Fecha de aprobación. • Fecha de entrega. • Validez. • Fecha de solicitud. 	
<p>Observaciones:</p> <ul style="list-style-type: none"> • Al guardar el formulario se envía un correo al usuario confirmando la creación de su solicitud, y la secretaria notificando la nueva solicitud. 	
<p>Interfaz:</p>	

Tabla 37. Historia de usuario 11. Fuente: elaboración propia.

Historia de Usuario	
Número: 11	Nombre: Editar solicitud de certificación de notas.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos de estimación: 0.3	Iteración asignada: 1
<p>Descripción: El usuario con rol de administrador accede a la opción de contenidos en el panel de administración al presionar el botón editar se debe mostrar el formulario para modificar los campos necesarios del trámite junto a los botones guardar y cancelar. Los datos del formulario son:</p> <ul style="list-style-type: none"> • Tipo de solicitud. • Motivo. • Validez. 	
<p>Observaciones:</p> <ul style="list-style-type: none"> • El formulario está disponible solo para usuarios con rol administrador. 	

Tabla 38. Historia de usuario 12. Fuente: elaboración propia.

Historia de Usuario	
Número: 12	Nombre: Eliminar solicitud de certificación de notas.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de estimación: 0.3	Iteración asignada: 1
<p>Descripción: El usuario con rol de administrador accede a la opción de contenidos en el panel de administración, selecciona el trámite que desea eliminar y al presionar el botón de eliminar se muestra un mensaje para verificar si ese es el que se quiere eliminar. Además, se muestran los botones de eliminar y cancelar respectivamente.</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> • El formulario está disponible solo para usuarios con rol administrador. 	

Tabla 39. Historia de usuario 13. Fuente: elaboración propia.

Historia de Usuario	
Número: 13	Nombre: Visualizar información de subsanación de errores

Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos de estimación: 0.4	Iteración asignada: 1
Descripción: El módulo debe permitir visualizar la información detallada del trámite subsanación de errores.	
Observaciones: <ul style="list-style-type: none"> ● Esta información es visible para todo tipo de usuario. ● El botón de “solicitar trámite” no está disponible para usuarios que no estén autenticados. ● El botón de “solicitar trámite” no está disponible para usuarios que no son graduados UCI. 	
Interfaz:	

Tabla 40. Historia de usuario 14. Fuente: elaboración propia.

Historia de Usuario	
Número: 14	Nombre: Crear solicitud de subsanación de errores.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos de estimación: 0.5	Iteración asignada: 1
Descripción: El usuario con rol usuario autenticado y graduado UCI al acceder al módulo debe visualizar el botón “solicitar trámite” al presionarlo aparecerá un formulario con los campos necesarios para crear el trámite. Debe poseer además los botones de guardar y cancelar respectivamente. Los datos del formulario son: <ul style="list-style-type: none"> ● Tipo de solicitud. ● Motivo. ● Estado. ● Fecha de aprobación. ● Fecha de entrega. ● Validez. ● Fecha de solicitud. 	
Observaciones:	

- Al guardar el formulario se envía un correo al usuario confirmando la creación de su solicitud, y la secretaria notificando la nueva solicitud.

Interfaz:

The screenshot shows a web form with the following elements:

- Tipo de Solicitud:** A dropdown menu with options: Entidad, Personal, Validez, Nacional, Internacional.
- Fecha de Solicitud:** A date picker showing 15/09/2023.
- Motivo:** A large empty text input field.
- Fecha de Aprobación:** A date picker showing 15/09/2023.
- Fecha de Aprobación:** A second date picker showing 15/09/2023.
- Buttons:** 'Cancelar' and 'Guardar' buttons at the bottom right.

Tabla 41. Historia de usuario 15. Fuente: elaboración propia.

Historia de Usuario	
Número: 15	Nombre: Editar solicitud de subsanación de errores.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos de estimación: 0.3	Iteración asignada: 1
<p>Descripción: El usuario con rol de administrador accede a la opción de contenidos en el panel de administración al presionar el botón editar se debe mostrar el formulario para modificar los campos necesarios del trámite junto a los botones guardar y cancelar.</p> <p>Los datos del formulario son:</p> <ul style="list-style-type: none"> • Tipo de solicitud. • Motivo. • Validez. 	
<p>Observaciones:</p> <ul style="list-style-type: none"> • El formulario está disponible solo para usuarios con rol administrador. 	

Tabla 42. Historia de usuario 16. Fuente: elaboración propia.

Historia de Usuario	
Número: 16	Nombre: Eliminar solicitud de subsanación de errores.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de estimación: 0.3	Iteración asignada: 1

Descripción: El usuario con rol de administrador accede a la opción de contenidos en el panel de administración, selecciona el trámite que desea eliminar y al presionar el botón de eliminar se muestra un mensaje para verificar si ese es el que se quiere eliminar. Además, se muestran los botones de eliminar y cancelar respectivamente.

Observaciones:

- El formulario está disponible solo para usuarios con rol administrador.

Tabla 43. Historia de usuario 18. Fuente: elaboración propia.

Historia de Usuario	
Número: 18	Nombre: Visualizar solicitudes de trámites docentes.
Prioridad en negocio: Alta	Riesgo en Desarrollo: Medio
Puntos de estimación: 1	Iteración asignada: 1
<p>Descripción: el módulo debe permitir a los usuarios autenticados y graduados UCI que accedan a su perfil mostrar un listado donde aparecerán los datos de las solicitudes realizadas por el usuario Se mostrarán los datos de:</p> <ul style="list-style-type: none"> ● Tipo de solicitud. ● Motivo. ● Estado. ● Fecha de aprobación. ● Fecha de entrega. ● Validez. ● Fecha de solicitud. 	
<p>Observaciones:</p> <ul style="list-style-type: none"> ● El formulario está disponible solo para usuarios autenticados, graduados UCI que accedan a su perfil de usuario. 	
<p>Interfaz:</p>	

Tarjetas CRC

Tabla 44. Tarjeta CRC 2. Fuente: elaboración propia.

Tarjeta CRC	
Clase	tramites_certifico_de_estudios_terminados
Responsabilidades	Colaboradores
crear_tramites_certifico_de_estudios_terminados () editar_tramites_certifico_de_estudios_terminados () eliminar_tramites_certifico_de_estudios_terminados () visualizar_tramites_certifico_de_estudios_terminados ()	notoficacion_correo tramites_sincronizacion

Tabla 45. Tarjeta CRC 3. Fuente: elaboración propia.

Tarjeta CRC	
Clase	tramites_subsananacion_de_errores
Responsabilidades	Colaboradores
crear_tramites_subsananacion_de_errores () editar_tramites_subsananacion_de_errores () eliminar_tramites_subsananacion_de_errores () visualizar_tramites_subsananacion_de_errores ()	notoficacion_correo tramites_sincronizacion

Tabla 46. Tarjeta CRC 4. Fuente: elaboración propia.

Tarjeta CRC	
Clase	tramites_certificacion_de_notas
Responsabilidades	Colaboradores
crear_tramites_certificacion_de_notas () editar_tramites_certificacion_de_notas () eliminar_tramites_certificacion_de_notas () visualizar_tramites_certificacion_de_notas ()	notoficacion_correo tramites_sincronizacion

```

// Un método para probar La función enviar_datos.
public function testEnviarDatosValido() {
    // Crear un mock de FormStateInterface con va
    $form_state = $this->createMockFormState([
        'uid' => [0 => ['target_id' => 1]],
        'field_tipo_de_solicitud' => [0 => ['target
        'field_validez' => [0 => ['value' => '1 año
        'field_descripcion' => [0 => ['value' => 'N
        'field_fecha_de_solicitud_' => [0 => ['valu
        'field_fecha_de_aprobacion' => [0 => ['valu
        'field_fecha_de_entrega' => [0 => ['value'
    ]]);

    // Crear un mock de User con un ci válido
    $user = $this->createMockUser('12345678');

    // Crear un mock de La función User::load
    $user_load = $this->getFunctionMock('Drupal\s
    $user_load->expects($this->once())->willReturn

```

Figura 23. Prueba unitaria testEnviarDatosValido (). Fuente: elaboración propia.

```

// Un método para probar La función admision_pre
public function testAdmisionPreprocessNodeAnonim
    // Crear un mock de AccountInterface con un id
    $current_user = $this->createMockCurrentUser(0

    // Crear un mock de La función \Drupal::curren
    $drupal_current_user = $this->getFunctionMock(
    $drupal_current_user->expects($this->once())->

    // Crear un array vacío para Las variables
    $variables = [];

    // Llamar a La función admision_preprocess_nod
    admision_preprocess_node($variables);

    // Verificar que Las variables se asignan corr
    $this->assertFalse($variables['logueado']);
    $this->assertFalse($variables['es_uci']);
}

```

Figura 24. Prueba unitaria función de validación {% if logueado and es_uci %}. Fuente: elaboración propia.

```
array(13) { ["id"]=> string(47) "tramites_notificaciones_tramites_notificaciones" ["module"]=> string(23) "tramites_notificaciones" ["key"]=> string(23) "tramites_notificaciones" ["to"]=> string(14) "ramonma@uci.cu" ["from"]=> string(14) "ramonma@uci.cu" ["reply-to"]=> NULL ["langcode"]=> string(2) "es" ["params"]=> array(2) { ["subject"]=> string(27) "Nueva solicitud de trámite" ["message"]=> object(Drupal Core StringTranslation:TranslatableMarkup)#60333 (5) { ["string":protected]=> string(467) "
```

Solicitud de trámites

Estimado @nombre:

Su solicitud del trámite @tramite ha sido enviada.

Atentamente:

Secretaria General de la UCI

Universidad de las Ciencias Informáticas

Información: www.uci.cu

Visite nuestra web: Universidad de las Ciencias Informáticas

```
" ["arguments":protected]=> array(2) { ["@nombre"]=> string(8) "aa aa aa" ["@tramite"]=> string(32) "Certifico de estudios terminados" } ["translatedMarkup":protected]=> string(467) "
```

Solicitud de trámites

Estimado @nombre:

Su solicitud del trámite @tramite ha sido enviada.

Atentamente:

Secretaria General de la UCI

Universidad de las Ciencias Informáticas

Información: www.uci.cu

Visite nuestra web: Universidad de las Ciencias Informáticas

Figura 25. Resultados de pruebas funcionales. Configuración de envío de correos. Fuente: elaboración propia.