

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

LUDOLAB CONNECTSYS: SISTEMA INFORMÁTICO PARA LA GESTIÓN DE EVENTOS DE FORMACIÓN Y SERVICIO A LA COMUNIDAD DEL LABORATORIO LUDOLAB

COMPONENTE ADMINISTRATIVO

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO/A EN
SOFTWARE**

LUIS GONZALO CHILIGUANO SUAREZ

`luis.chiliguano@epn.edu.ec`

DIRECTOR: VICTOR VICENTE VELEPUCHA BONETT

`victor.velepucha@epn.edu.ec`

DQM, JULIO 2024

CERTIFICACIONES

Yo, LUIS GONZALO CHILIGUANO SUAREZ declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

LUIS GONZALO CHILIGUANO SUAREZ

Certifico que el presente trabajo de integración curricular fue desarrollado por LUIS GONZALO CHILIGUANO SUAREZ , bajo mi supervisión.

VICTOR VICENTE VELEPUCHA BONETT
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

LUIS GONZALO CHILIGUANO SUAREZ

VICTOR VICENTE VELEPUCHA BONETT

GINNO SEBASTIAN TAIMAL VILLARROEL

MIGUEL ANGEL MUENALA QUINGA

DEDICATORIA

Dedico este trabajo a mis padres, quienes me han brindado todo lo necesario para llegar hasta aquí.

A mi madre, Natalia Suárez, una mujer fuerte y valiente que ha enfrentado el desafío de criarnos con determinación, llevándonos a alcanzar nuestras metas y siendo una parte fundamental de cada logro.

A mi padre, Luis Arturo Chiliguano, quien ha trabajado incansablemente para proporcionarnos el presente que disfrutamos hoy.

A mi hermano David, quien ha crecido a mi lado y me ha acompañado en cada etapa de mi vida. Espero ser un ejemplo e inspiración para él, y que pueda completar sus estudios y alcanzar sus propias metas.

A mi hermana Erika, cuyo apoyo incondicional ha sido una constante, incluso en los momentos difíciles para nuestros padres.

AGRADECIMIENTO

Agradezco profundamente a mis padres, Luis y Natalia, quienes han dado todo de sí para que pudiera culminar esta carrera. También a mi hermana, por su apoyo constante, especialmente en los momentos difíciles; y a mi hermano, quien ha estado presente en cada etapa de mi vida. Así también, extendo mi gratitud a toda mi familia, quienes en algún momento del trayecto me brindaron su ayuda.

Agradezco a mis amigos, quienes formaron parte de esta etapa en la universidad e hicieron más llevaderas las horas de trabajo, esfuerzo y estrés que compartimos.

A mis compañeros en la realización de este trabajo, Ginno, quien me incluyó en esta idea, y Miguel, quien siempre estuvo dispuesto a colaborar para obtener los resultados que esperábamos, les doy mis más sinceros agradecimientos.

Extendo mi gratitud al MSc. Víctor Velepucha, quien me guió a lo largo del desarrollo de este trabajo, y al Dr. Marco Santorum, quien estuvo presente en la concepción inicial de este proyecto. Y a cada uno de mis maestros a lo largo de la carrera, les doy mi más sincero agradecimiento.

Finalmente, y sobre todo, doy las gracias a Dios, en quien encontré fortaleza en los momentos difíciles y quien, por su perfecta voluntad, ha permitido que todo esto sea posible.

Índice general

1. INTRODUCCIÓN	1
1.1. Objetivos	3
1.1.1. Objetivo general	3
1.1.2. Objetivos específicos	3
1.2. Alcance	4
1.3. Marco teórico	4
1.3.1. Gestión académica y estudiantil	4
1.3.2. Sistema de administración estudiantil	5
1.3.3. Single Page Application (SPA)	5
1.3.4. Kanban	6
1.3.5. Kanban en el ámbito del desarrollo de software	7
2. METODOLOGÍA	10
2.1. Metodología Kanban	10
2.1.1. Principios directores de Kanban	11
2.1.2. Principios de despliegue de servicios	11
2.1.3. Valores de Kanban	11
2.1.4. Roles en Kanban	12
2.1.5. Prácticas de Kanban	12
2.2. Herramientas para el desarrollo	14
2.3. Desarrollo	18
2.3.1. Planificación y análisis	18
2.3.2. Diseño	25
2.3.3. Implementación	29

3. EVALUACIÓN Y RESULTADOS	60
3.1. Pruebas de funcionalidad	60
3.2. Pruebas de usabilidad	65
4. CONCLUSIONES Y RECOMENDACIONES	71
4.1. Conclusiones	71
4.2. Recomendaciones	72
5. REFERENCIAS BIBLIOGRÁFICAS	74
ANEXOS	78

Índice de figuras

1.1. Reglas de Kanban	7
1.2. Características de Kanban	8
1.3. Tablero Kanban básico	9
2.1. Proceso general de Inclusión Digital	20
2.2. Tablero Kanban inicial	23
2.3. Diagrama de base de datos para el componente administrativo desarrollado	26
2.4. Arquitectura de la aplicación	27
2.5. Mockup para la interfaz de creación de cursos	28
2.6. Mockup para la interfaz de lista de cursos	28
2.7. Mockup para la interfaz de registro de estudiantes	29
2.8. Estructura de proyecto frontend general	30
2.9. Organización de interfaces por áreas (roles)	30
2.10. Estructura proyecto backend	31
2.11. Estructura biblioteca de archivos compartidos	32
2.12. Tablero Kanban inicial para el módulo de Estudiantes	33
2.13. Cambio al estado de En Progreso de la HU01 en el tablero Kanban	33
2.14. Formulario de registro de estudiante	34
2.15. Pantalla de información detallada de un estudiante	35
2.16. Tablero Kanban al completar la programación de las historias de usuario HU01 y HU03	36
2.17. Tablero Kanban al iniciar las pruebas para las historias de usuario HU01 y HU03	36
2.18. Tablero Kanban al iniciar la integración de las funcionalidades descritas en HU01 y HU03	37
2.19. Tablero Kanban al iniciar la implementación del submódulo de Instructores	38

2.20. Formulario de registro de instructores	39
2.21. Pantalla de información detallada de un instructor	40
2.22. Tablero Kanban final para el submódulo de instructores	41
2.23. Tablero Kanban inicial para el submódulo de Cursos	42
2.24. Formulario para la creación de un nuevo curso	42
2.25. Formulario para la creación de un nuevo periodo de curso	43
2.26. Información y gestión de cursos en progreso	43
2.27. Pantalla de información y gestión de cursos	44
2.28. Pantalla de edición de la información general de un curso	44
2.29. Lista de periodos por curso	45
2.30. Pantalla de edición de un periodo	45
2.31. Tablero Kanban para implementar requerimientos de grupos	46
2.32. Parámetros para la creación automática de grupos	47
2.33. Resultados de emparejamiento y creación de grupos	48
2.34. Asignación manual de estudiantes e instructores para la creación de grupos	49
2.35. Tablero Kanban tras la implementación del submódulo de cursos	49
2.36. Tablero kanban inicial para el submódulo de repositorio	50
2.37. Dependencias necesarias para la interfaz de repositorio	51
2.38. Lógica de autenticación	51
2.39. Interfaz inicial de repositorio	52
2.40. Interfaz para subir archivo	52
2.41. Lógica para la selección de archivos	53
2.42. Selección de elemento a subir	53
2.43. Método para subir archivos a Sharepoint	54
2.44. Resultado de carga exitosa	55
2.45. Carga de archivo exitoso en el repositorio de Sharepoint	55
2.46. Tablero Kanban tras la implementación del submódulo de repositorio de do- cumentos	56
2.47. Cursos disponibles para inscripción	57
2.48. Ingreso disponibilidad estudiante	57
2.49. Cursos disponibles para registro como instructor	58
2.50. Ingreso de disponibilidad de un instructor	58
2.51. Tablero Kanban tras finalizar la implementación del componente administrati- vo en su totalidad	59

3.1. Caso de prueba para el registro exitoso de estudiantes	61
3.2. Registro exitoso de un estudiante	61
3.3. Caso de prueba para el registro exitoso de instructores	62
3.4. Registro exitoso de un instructor	62
3.5. Caso de prueba para la creación exitosa de un curso	63
3.6. Creación exitosa de curso	63
3.7. Curso creado en listado de cursos	64
3.8. Caso de prueba para la carga exitosa de documentos al repositorio	64
3.9. Preguntas del cuestionario de usabilidad del sistema SUS	69
3.10. Rangos de puntuación SUS	70

Índice de Tablas

2.1. Versiones del entorno .NET	15
2.2. Roles en el proyecto de inclusión digital	19
2.3. Historias de usuario para el módulo administrativo	21
3.1. Detalle de tareas y subtareas para el rol de administrador	66
3.2. Detalle de tareas y subtareas para el rol de estudiante	68
3.3. Detalle de tareas y subtareas para el rol de instructor	68

RESUMEN

El presente trabajo de titulación expone el desarrollo de un módulo de administración para el sistema informático para la gestión de eventos de formación y servicio a la comunidad del laboratorio LudoLab, el cual está destinado a controlar y gestionar los cursos y participantes de los programas de capacitación que se ofrecen en el laboratorio. El objetivo general de su desarrollo fue agilizar los procesos académicos, la administración de la información de los participantes y la supervisión del trabajo de los instructores. A través del uso de la metodología ágil Kanban, se implementaron submódulos específicos para la creación y configuración de cursos, la inscripción y matriculación de estudiantes, y el registro y gestión de la disponibilidad horaria de los instructores voluntarios. Adicionalmente, se desarrolló un submódulo para archivar la información del trabajo realizado durante los cursos a modo de repositorio de documentos. Las pruebas de funcionalidad, implementadas a través de casos de prueba, confirmaron que el sistema cumple con los requisitos establecidos. Por otro lado, la evaluación de usabilidad mediante una encuesta SUS (Escala de Usabilidad del Sistema) arrojó un puntaje de usabilidad medio de 73,67 puntos; esto indicó una percepción satisfactoria por parte de los usuarios. Las conclusiones destacan la efectividad del módulo en mejorar la organización académica y administrativa, mientras que las recomendaciones sugieren la continua mejora del sistema y la capacitación de los usuarios.

PALABRAS CLAVE - Sistema de administración estudiantil, Kanban, gestión académica, capacitación, optimización de procesos

ABSTRACT

This thesis presents the development of an administration module for the information system for managing training events and community service at the LudoLab laboratory. This module is designed to control and manage the courses and participants of the training programs offered in the laboratory. The main objective of its development was to streamline academic processes, the administration of participant information, and the supervision of instructor work. Through the use of the agile Kanban methodology, specific submodules were implemented for the creation and configuration of courses, the enrollment and registration of students, and the registration and management of volunteer instructors' availability. Additionally, a submodule was developed to archive the information of the work carried out during the courses, functioning as a document repository. The functionality tests, implemented through test cases, confirmed that the system meets the established requirements. Moreover, the usability evaluation using a SUS (System Usability Scale) survey yielded an average usability score of 73.67 points, indicating a satisfactory perception by users. The conclusions highlight the effectiveness of the module in improving academic and administrative organization, while the recommendations suggest the continuous improvement of the system and the training of users.

KEYWORDS - Student administration system, Kanban, academic management, training, process optimization

Capítulo 1

INTRODUCCIÓN

Dentro de la Escuela Politécnica Nacional, se ha promovido la creación de unidades académicas y departamentos dedicados a impulsar y profundizar en distintos dominios científicos y tecnológicos. Estas entidades son fundamentales para la conducción de proyectos de índole educativo, de investigación y de servicio a la comunidad; facilitando de esta forma el avance y la aplicación del conocimiento en beneficio de la sociedad.

Bajo este contexto, el Laboratorio de Investigación de Sistemas de Información e Inclusión digital LudoLab, establecido en 2019 por la Escuela Politécnica Nacional, ha asumido un papel protagonista en la mitigación de la brecha digital a través del proyecto Inclusión Digital. Este se caracteriza por ofrecer clases personalizadas llevadas a cabo por estudiantes voluntarios de la EPN, quienes cumplen el rol de instructores. El objetivo principal del proyecto de inclusión es brindar competencias digitales fundamentales a sectores menos representados, entre ellos docentes de escuelas rurales y personas adultas mayores, para facilitar su acceso a las oportunidades que ofrece el entorno digital moderno.

En el ámbito particular de los eventos de capacitación de LudoLab, la gestión de la información constituye un terreno complejo y en constante incremento, esto debido a la generación de una gran cantidad de datos, que incluyen registros de inscripciones, creación de cursos, calificaciones, encuestas de satisfacción, entre otros. De modo que estos datos se han ido gestionando con herramientas ofimáticas no especializadas o de forma manual, lo que dificulta en gran medida la obtención de información valiosa para mejorar los programas académicos.

Por otro lado, es notable que en los últimos años se ha venido llevando a cabo un gran proceso de transformación digital en diferentes sectores, siendo el educativo uno de los campos que más oportunidades presenta [1]. Es así que un gran porcentaje de institucio-

nes educativas han decidido optimizar sus procesos académicos a través del uso de diferentes herramientas tecnológicas, teniendo como uno de sus objetivos principales mejorar el manejo de la información.

De esta manera, han surgido los Sistemas de Administración Estudiantil, que son plataformas diseñadas para facilitar la gestión de la información relacionada con los estudiantes y los programas académicos, permitiendo automatizar procesos como matriculaciones, gestión de horarios, y seguimiento académico; mejorando así la eficiencia operativa y reduciendo la carga administrativa de las instituciones [2].

Siguiendo esta tendencia de transformación digital, y buscando superar los desafíos presentes en la gestión de datos de los programas de capacitación de LudoLab, se propone desarrollar un módulo administrativo que haga parte de un sistema estudiantil especializado, de forma que facilite el manejo eficiente de la información generada en el transcurso de los eventos de capacitación que ofrece el laboratorio. Este módulo de administración estudiantil incluirá varias funcionalidades enfocadas en mejorar la organización y eficiencia de los programas de capacitación.

En primera instancia, se ha contemplado la elaboración de un submódulo que permita la creación y administración de los cursos que surjan como parte de los programas de capacitación. Así, se facilitará su concepción y, organización en periodos y grupos o paralelos. Además, será posible asignar un número determinado de estudiantes e instructores a estos grupos, para asegurar una distribución personalizada de carga estudiantil por instructor.

En cuanto a la gestión de participantes, se ha considerado un submódulo que permita el registro de nuevos estudiantes, su matriculación en los cursos y su asignación a los paralelos correspondientes. De esta forma, se otorga la capacidad de obtener información histórica de quienes han realizado un curso de capacitación en periodos pasados y vigentes, además de mantener un registro actualizado y preciso de todos los estudiantes dentro del sistema.

Por otro lado, la gestión de instructores se centrará en su registro y asignación a los cursos y grupos correspondientes. En este sentido, se asegurará que cada grupo de estudiantes, independientemente de su cantidad, cuente con un instructor adecuado a sus necesidades educativas.

Además, se ha contemplado la capacidad de subir archivos a un sitio de SharePoint que cumplirá el trabajo de un repositorio de archivos, así se garantizará que se provean las evidencias del progreso de los estudiantes y el trabajo realizado de los instructores. Además, el repositorio asegurará que todos los documentos se encuentren centralizados y sean

fácilmente accesibles para los usuarios autorizados en el sistema.

Con estas características en mente, se busca alcanzar los siguientes objetivos.

1.1. Objetivos

1.1.1. Objetivo general

Desarrollar un módulo de administración que permita controlar y gestionar cursos y participantes de los programas de capacitación de LudoLab, con el fin de agilizar los procesos académicos, la administración de la información de los estudiantes y la supervisión del trabajo de los instructores.

1.1.2. Objetivos específicos

1. Aplicar un enfoque basado en una metodología ágil adecuada en cada fase del ciclo de vida de desarrollo del software para asegurar una gestión eficiente y adaptativa del proyecto.
2. Implementar un submódulo que facilite la creación y configuración de cursos en la plataforma, desarrollando interfaces de fácil uso, para que los administradores puedan realizar dichos procesos de manera asequible.
3. Implementar un submódulo que agilice el proceso de inscripción y matriculación de estudiantes en los cursos de capacitación de LudoLab mediante el uso de formularios simplificados, con el fin de optimizar la administración de los datos estudiantiles y mejorar la eficiencia en la gestión académica.
4. Implementar un submódulo que permita el registro de instructores y agilice la gestión de su disponibilidad horaria para mejorar la organización de las actividades académicas, haciendo más intuitivos y fáciles los procesos de administración de los datos de los instructores voluntarios.
5. Implementar un submódulo para archivar la información que respalde el trabajo realizado durante los cursos (repositorio) y así mantener un registro organizado.
6. Evaluar el sistema en cuanto a funcionalidad y usabilidad, realizando las pruebas necesarias y recogiendo retroalimentación detallada, para asegurar que el sistema cumpla con los requisitos funcionales y sea de fácil uso.

1.2. Alcance

El módulo de administración contempla una fase previa de diseño de interfaces, cuyo objetivo es simplificar el ingreso de datos para los administradores; además, a través de estas se buscará poner a su alcance las herramientas necesarias para agilizar los procesos administrativos de creación de cursos, registro de participantes y gestión de datos.

El desarrollo de este módulo se llevará a cabo utilizando una metodología ágil, con el objetivo de asegurar una mayor flexibilidad en la gestión del proyecto. Este enfoque permitirá una rápida adaptación ante posibles cambios, facilitando la realización de ajustes y mejoras. Así también, al adoptar una metodología ágil, no solo se busca mejorar la eficiencia del proceso de desarrollo, sino también aumentar el valor del producto ante los ojos de sus beneficiarios.

Es importante destacar que el desarrollo de este componente, junto con los otros componentes implementados de forma simultánea para el mismo sistema, no representa una solución definitiva e inamovible para la gestión de la información académica y estudiantil de los cursos de capacitación impartidos en LudoLab. Por el contrario, se trata de una primera aproximación que puede servir como base para futuras mejoras o la adición de nuevas funcionalidades.

1.3. Marco teórico

1.3.1. Gestión académica y estudiantil

Cuando se habla de gestión académica se hace referencia a las actividades orientadas a mejorar los proyectos de una institución educativa y sus procesos pedagógicos, con la finalidad de responder de forma adecuada a las necesidades académicas presentes en su entorno [3]. Por lo general, la gestión académica incluye la elaboración de los planes de estudio, la administración del personal académico, la matrícula y el seguimiento del progreso de los estudiantes, entre otros.

Uno de los aspectos más importantes de la gestión académica es la planificación y organización de los recursos educativos. Así, una gestión académica eficiente busca optimizar el uso de estos recursos y garantizar que los estudiantes reciban una educación de alta calidad, adaptada a las demandas en constante cambio dentro del contexto educativo [4]. En este sentido, la tecnología desempeña un papel crucial al proporcionar herramientas que

facilitan la automatización y mejora de estos procesos, permitiendo una administración más ágil y precisa [5].

1.3.2. Sistema de administración estudiantil

También conocido como sistema de información estudiantil (SIS), un sistema de administración estudiantil (SAE) puede definirse en líneas generales como el marco digital que permite a las instituciones educativas gestionar los datos de los estudiantes de forma integral [6]. Por lo general, estos sistemas tienen funcionalidades que simplifican los procesos académicos de inscripción, programación de cursos, seguimiento del rendimiento académico, entre otros.

El objetivo principal de un SAE es permitir una mejor organización y facilitar el acceso a la información, lo que mejora la eficiencia operativa de las instituciones, pues al centralizar y automatizar tareas administrativas, los educadores y el personal administrativo pueden dedicar más tiempo a actividades pedagógicas y de apoyo a los estudiantes. Por otro lado, la implementación de un SAE también contribuye a una toma de decisiones más informada, ya que los datos recopilados y gestionados con estos sistemas pueden analizarse para identificar tendencias y áreas de mejora [7].

Adicionalmente, un SAE puede mejorar la precisión de los datos y la comunicación entre los diferentes actores involucrados en los procesos educativos, pues proporciona una fuente única y precisa de información, lo que reduce las discrepancias y errores en los datos; por lo cual, se mejora de esta forma la eficiencia administrativa [8].

Por estas características, la adopción y correcto uso de un SAE han llegado a representar las razones fundamentales que guían al éxito y a la modernización a las instituciones educativas en la concurrente era digital [9]. Luego, no resulta nada extraño que se haya producido un cambio sísmico en los sistemas educativos durante los últimos años, pues las instituciones buscan alcanzar una ansiada digitalización y automatización de los procesos [10]

1.3.3. Single Page Application (SPA)

Una aplicación de página única (SPA) es una variante de aplicación web que carga un solo documento web y actualiza dinámicamente el contenido de su cuerpo a medida que el usuario interactúa con la aplicación [11]. Es decir, en lugar de cargar varias páginas enteras desde el servidor para cada interacción, una SPA hace una sola carga y luego se ocupa

de modificar la interfaz del usuario en respuesta a sus acciones. Este enfoque proporciona una experiencia de usuario más rápida y fluida, buscando acercarse a la de una aplicación de escritorio. Si bien los SPAs suelen ser construidos principalmente mediante el uso de marcos de trabajo basados completamente en JavaScript, como React, Angular o Vue.js; existe una alternativa creada por Microsoft que permite su desarrollo utilizando C# como lenguaje principal. Esta alternativa, conocida como Blazor, facilita a los desarrolladores especializados en C# la creación de SPAs de manera sencilla, aprovechando su conocimiento en este lenguaje y el ecosistema .NET. Más adelante se profundizará en la descripción de Blazor dentro de la sección herramientas de desarrollo.

1.3.4. Kanban

Kanban como metodología de producción tiene sus orígenes en la industria manufacturera automotriz japonesa en la década de 1940. Fue desarrollada por Taiichi Ohno como parte del sistema de producción de Toyota, el cual estaba centrado en mejorar la eficiencia del trabajo y reducir los desperdicios a través del sistema Just-In-Time. De esta forma, el fabricante evitaba la acumulación de material en el inventario y, en su lugar, empleaba tarjetas a modo de señales visuales para comunicar la deficiencia de materiales específicos, adquiriendo de esta forma únicamente lo necesario, lo cual aseguraba un flujo de trabajo continuo [12]. Por ello, Kanban está basado en el uso de señales visuales para gestionar el esfuerzo y la organización del trabajo de un equipo, facilitando de esta forma la identificación de cuellos de botella en el flujo de trabajo, lo cual permite mejorar la calidad del proceso y, por extensión, la calidad del producto final. Kanban promueve los siguientes principios [13]:

- Calidad desde el principio: Enfocarse en hacer las cosas bien desde el inicio para evitar arreglos posteriores que consumen más tiempo.
- Evitar desperdicios: Realizar solo lo necesario, evitando tareas innecesarias siguiendo el principio YAGNI (You Aren't Gonna Need It - No lo necesitarás).
- Mejora continua: Mejorar permanentemente los desarrollos según los objetivos.
- Flexibilidad: Priorizar y ajustar tareas según las necesidades del momento.
- Relaciones con proveedores: Mantener relaciones a largo plazo con los proveedores.

Además, se consideran seis reglas principales para su correcta adopción, las cuales se representan en la Figura 1.1:

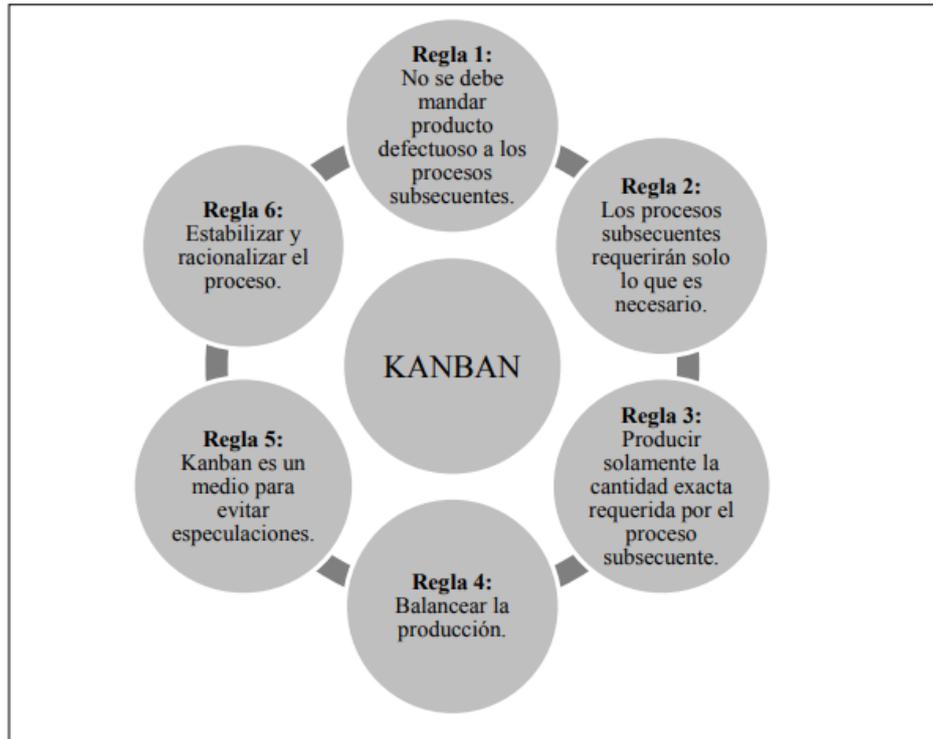


Figura 1.1: Reglas de Kanban

[14]

1.3.5. Kanban en el ámbito del desarrollo de software

La metodología Kanban tiene sus primeras apariciones en el espectro del desarrollo de software en el año 2004, cuando David J. Anderson adaptó estos principios para ayudar a un grupo de desarrolladores de Microsoft a mejorar su funcionamiento y organización. La misma se popularizó tras el lanzamiento del libro “Kanban: Successful Evolutionary Change for Your Technology Business” de autoría del propio Anderson [15].

En este contexto, la metodología Kanban se implementa en base a la visualización de tareas en un tablero, estableciendo límites de trabajo en progreso en cada etapa y fomentando la mejora continua. A continuación, se detallan las principales características de Kanban como metodología de desarrollo de software en la Figura 1.2.

Característica	Descripción
Visualiza el flujo de trabajo	Divide el trabajo en bloques, escribe cada elemento en una tarjeta y se coloca en el tablero. Utiliza columnas con nombre para ilustrar dónde está cada elemento en el flujo de trabajo.
Limita el trabajo en curso (WIP)	Asigna límites concretos al número de elementos que pueden estar en progreso en cada estado del flujo de trabajo.
Mide el tiempo de ciclo (lead time) medio para completar un elemento	Optimiza el proceso para que el lead time sea tan pequeño y predecible como sea posible.

Figura 1.2: Características de Kanban

[16]

El núcleo de la metodología Kanban es, sin duda, el tablero Kanban. Este se organiza en columnas que muestran el estado del trabajo asignado a cada miembro del equipo mediante tarjetas que detallan sus actividades. Al proporcionar una vista completa del flujo de trabajo, el tablero ayuda a identificar cuellos de botella y permite al equipo detectar problemas que bloquean el progreso, facilitando así una retroalimentación continua y un alto nivel de transparencia [16]. Las tarjetas Kanban representan visualmente las tareas individuales del equipo y los límites de trabajo en progreso, los cuales crean un sistema de arrastre. De este modo, el trabajo se “arrastra” al flujo cuando otro trabajo se completa y queda capacidad disponible, en lugar de “empujar” trabajos al sistema con nuevas demandas (Just in time) [17]. Esto asegura un seguimiento eficiente y una gestión optimizada del trabajo. En su forma básica, un tablero Kanban tiene tres columnas que indican el estado de una tarea o actividad: por hacer (To Do), en proceso (Doing) y terminado (Done), como se muestra en la Figura 1.3.



Figura 1.3: Tablero Kanban básico

Cabe recalcar que el tablero Kanban se puede adaptar según las necesidades específicas del equipo que lo utiliza. En el desarrollo de software, el tablero debe proporcionar una visión clara del proceso, por lo que las columnas o estados del tablero pueden alinearse con las fases del ciclo de vida del desarrollo de software, aunque no es estrictamente necesario que haya una correspondencia exacta para cada fase.

Capítulo 2

METODOLOGÍA

Para llevar a cabo el desarrollo del proyecto, se decidió utilizar la metodología Kanban debido a su idoneidad para equipos pequeños. Como se explicó anteriormente en la sección de marco teórico 1.3, Kanban proporciona una manera flexible y sencilla de organizar y priorizar el trabajo, lo cual es esencial dada la necesidad de adaptabilidad y eficiencia del equipo. A diferencia de otros marcos de trabajo como Scrum, que requieren roles específicos y ceremonias formales, Kanban permite una gestión del trabajo más fluida y menos estructurada. Esto se adapta mejor a las limitaciones del equipo y del contexto en el que se desarrolla este trabajo. Además, la capacidad de visualizar todas las tareas en un tablero kanban ayuda a gestionar la carga de trabajo de manera efectiva mediante la implementación de límites en el trabajo en progreso. De esta forma, se asegura que el equipo pueda mantener un flujo constante de trabajo y responder ágilmente a posibles cambios, maximizando la productividad y la entrega continua de valor a los potenciales beneficiarios del sistema.

2.1. Metodología Kanban

Como se mencionó anteriormente, la metodología Kanban permite definir, gestionar y mejorar servicios que implican la entrega de trabajo del conocimiento, como el desarrollo de software, y se basa en hacer visible el proceso de producción para asegurar que el servicio funcione de manera óptima y con la cantidad adecuada de trabajo [17]. Si bien Kanban es poco prescriptivo en comparación con otras metodologías y marcos de trabajo ágiles, el “padre” de Kanban, David Anderson, define algunos aspectos a tener en cuenta para su implementación [15].

2.1.1. Principios directores de Kanban

Kanban cuenta con tres principios directores que se basan en las necesidades de la organización que busca implementarlo:

- **Sostenibilidad:** Mantener un ritmo de trabajo sostenible y enfocarse en la mejora continua.
- **Orientación al Servicio:** Enfocarse en proporcionar servicios que cumplan con las expectativas del cliente.
- **Supervivencia:** Asegurar que la organización pueda adaptarse y prosperar en un entorno cambiante.

2.1.2. Principios de despliegue de servicios

Adicionalmente a los principios que dirigen la implementación de la metodología Kanban, Carmichael [17] define tres principios en cuanto a la entrega o despliegue de servicios, como lo es en el caso de desarrollo de software:

- **Comprender las necesidades y expectativas de los clientes y centrarse en ellas.**
- **Administrar el trabajo de forma en la que se permita que las personas se auto organicen en torno a sus tareas designadas.**
- **Adaptar las políticas internas para mejorar los resultados tanto para el cliente como para el negocio.**

2.1.3. Valores de Kanban

Kanban está guiado por una serie de valores que son fundamentales para su implementación efectiva [17]. Estos valores incluyen:

- **Transparencia:** Compartir información abiertamente mejora el flujo de valor del negocio.
- **Equilibrio:** Mantener un balance entre diferentes aspectos y capacidades para lograr efectividad.
- **Colaboración:** Trabajar juntos para mejorar la manera en que las personas colaboran.

- Foco en el Cliente: Centrar el sistema en la entrega de valor al cliente.
- Flujo: Asegurar que el trabajo fluya de manera continua y predecible.
- Liderazgo: Inspirar a otros a través del ejemplo y la reflexión.
- Entendimiento: Conocer tanto individual como organizativamente el punto de partida para mejorar.
- Acuerdo: Compromiso conjunto hacia los objetivos, respetando las diferencias.
- Respeto: Valorar y considerar a las personas en el proceso.

2.1.4. Roles en Kanban

Anderson explica que no existen roles obligatorios en Kanban, y que la metodología no crea nuevas posiciones dentro de la organización. Sin embargo, a lo largo de su uso, han surgido dos roles que pueden integrarse a la metodología en su implementación [17].

- Gestor de Peticiones de Servicio (Service Request Manager): Responsable de entender las necesidades del cliente y gestionar las peticiones de trabajo.
- Gestor de Prestación de Servicio (Service Delivery Manager): Responsable de asegurar que el trabajo fluya y se entregue según lo prometido.

Cabe recalcar la importancia de no centrarse en asignar títulos formales e inamovibles, sino en comprender el propósito de estos roles. Por lo tanto, es útil pensar en estos roles como momentáneos, de modo que las personas pueden cumplirlos cuando deban desempeñar funciones específicas.

2.1.5. Prácticas de Kanban

Las prácticas de la metodología Kanban permiten guiar la forma en la que se utiliza un sistema kanban. Existen seis prácticas principales [17], las cuales se describen a continuación:

1. Visualizar

En Kanban, la visualización implica el uso de un tablero que represente el flujo de trabajo. Este tablero, que puede ser físico o digital, muestra las diferentes etapas del

proceso, desde el inicio hasta la entrega. Su principal función es facilitar la identificación de cuellos de botella y áreas de mejora, por lo que no existen restricciones para su diseño.

2. Limitar el trabajo en progreso (WIP)

Establecer límites en la cantidad de trabajo en progreso (WIP, por sus siglas en inglés Work in Progress) es un principio crucial en la metodología Kanban. Al limitar el WIP, se evita que el equipo se sobrecargue con demasiadas tareas simultáneas, lo que puede llevar a la pérdida de eficiencia y calidad. Establecer un número controlado de tareas en progreso ayuda a mantener un flujo constante y predecible de trabajo, asegurando que las tareas se completen de manera regular y sin interrupciones.

3. Gestionar el flujo

La gestión del flujo implica supervisar y optimizar cómo se mueve el trabajo a través del proceso. Al gestionar el flujo, es fundamental monitorear continuamente el proceso para asegurarse de que el trabajo fluya sin interrupciones. La identificación y eliminación de bloqueos y cuellos de botella ayuda a mejorar la entrega de valor al cliente y asegura que el equipo pueda trabajar de manera más efectiva. Además, una buena gestión del flujo permite hacer ajustes basados en datos reales, lo que facilita la toma de decisiones informadas.

4. Hacer explícitas las políticas

Es esencial que las políticas que rigen el sistema sean claras y bien comunicadas a todo el equipo. Definir criterios de entrada y salida para cada etapa del proceso asegura que todos los miembros comprendan y sigan las mismas reglas. De esta manera, no solo se facilita la mejora continua, sino que también es posible asegurar la coherencia en el trabajo del equipo. Por ejemplo, una política podría especificar cuándo una tarea puede pasar de “en progreso” a “completada” o definir los requisitos para que una tarea sea considerada terminada. Al tener estas reglas claramente definidas y comunicadas, se reduce la ambigüedad y se mejora la eficiencia y la calidad del trabajo.

5. Implementar circuitos de retroalimentación

Establecer circuitos de retroalimentación regulares es fundamental en Kanban. Estos pueden ser reuniones periódicas, como reuniones diarias, revisiones de trabajo y retrospectivas, que permiten al equipo comunicarse de manera continua y efectiva.

Las reuniones ayudan a identificar problemas rápidamente y encontrar soluciones de inmediato, mejorando el rendimiento general del equipo, mientras que la retroalimentación regular facilita la evolución del proceso y la mejora continua, pues proporciona un espacio para reflexionar sobre lo que está funcionando y lo que no, y permite hacer ajustes en consecuencia.

6. Mejorar colaborativamente, evolucionar experimentalmente

Kanban promueve la mejora continua mediante la colaboración y la experimentación. Fomentar un enfoque evolutivo permite al equipo adaptarse a los cambios y mejorar los procesos de manera constante. La experimentación controlada ayuda a descubrir prácticas más eficaces, lo cual beneficia al equipo y al flujo de trabajo en general, permitiendo una evolución continua basada en la experiencia y el aprendizaje.

2.2. Herramientas para el desarrollo

.NET

Se trata de una plataforma de desarrollo gratuita, multiplataforma y de código abierto creada por Microsoft y diseñada para compilar una variedad de aplicaciones [18]. La creación de aplicaciones robustas y escalables se posibilita con las herramientas y bibliotecas proporcionadas por esta plataforma de desarrollo. Adicionalmente, cuenta con soporte para numerosos lenguajes de programación, siendo C# el más popular. Además, el apartado de seguridad se cubre completamente en cada componente que hace parte de .NET.

Existen diferentes versiones de .NET, cada una con un enfoque específico, por lo que es importante diferenciar entre dos tipos muy parecidos, como lo muestra la Tabla 2.1:

Tabla 2.1: Versiones del entorno .NET

Versión	Descripción
.NET Framework	.NET original. Plataforma exclusiva para Windows y pensada para aplicaciones tradicionales, tanto de escritorio como web. Cuenta con mantenimiento activo, pero no se ha considerado como el estándar para futuros lanzamientos por parte de Microsoft.
.NET (Core)	.NET moderno. Es una implementación multiplataforma y de código abierto (compatible con macOS y Linux) ideal para construir aplicaciones de vanguardia. Microsoft recomienda su utilización debido a su vigencia en cuanto a actualizaciones regulares y mejoras continuas.

[18]

ASP.NET Core

ASP.NET Core es un marco multiplataforma moderno, de código abierto, diseñado para crear aplicaciones web. Construido sobre .NET Core, facilita a los desarrolladores crear APIs RESTful, servicios web y aplicaciones web ágiles. Se sabe que ASP.NET Core es compatible con varios sistemas operativos, incluidos Windows, Linux y macOS, lo que permite la creación de aplicaciones compatibles con la mayoría de los entornos sin que eso represente un mayor esfuerzo. Además, ASP.NET Core se integra perfectamente con tecnologías de vanguardia, lo que lo convierte en una de las mejores plataformas para desarrollar aplicaciones altamente disponibles y escalables. [19]. Para el desarrollo de este componente, se utilizó la versión .NET 8.

ASP.NET Core Blazor

Blazor es un marco web de front-end del ecosistema de ASP.NET Core que permite a los desarrolladores crear aplicaciones web interactivas utilizando el lenguaje C# en lugar del tradicional JavaScript. Este marco se basa en el uso de componentes web, elementos reutilizables de la interfaz de usuario que se ensamblan para construir aplicaciones complejas.

Blazor ofrece dos modelos de hospedaje: Blazor Server y Blazor WebAssembly. Por su parte, Blazor Server ejecuta las aplicaciones en el servidor y utiliza la biblioteca SignalR para la comunicación en tiempo real con el cliente, mientras que Blazor WebAssembly permite la ejecución de aplicaciones directamente en el navegador del cliente utilizando WebAssembly.

Esta flexibilidad facilita la creación de aplicaciones web ricas y de alto rendimiento, aprovechando todo el ecosistema y las herramientas disponibles en .NET. De esta forma, el uso de Blazor simplifica el desarrollo de aplicaciones de una sola página (SPA) y mejora la experiencia de creación al permitir la utilización de C# en todo el stack de desarrollo [20].

SQL Server

SQL Server es un sistema de gestión de bases de datos relacional desarrollado por Microsoft, el cual está construida para el manejo de grandes volúmenes de datos, proporcionando un acceso rápido y seguro a la información. La plataforma es reconocida por su facilidad de uso y administración gracias a SQL Server Management Studio (SSMS), pues ofrece una interfaz intuitiva que simplifica la gestión de bases de datos, copias de seguridad, monitorización del rendimiento y muchas otras características [21]. Para el desarrollo de este componente, se utilizó la version SQL Server 2022 Developer Edition.

Visual Studio

Visual Studio es un entorno de desarrollo de software, también conocido como IDE, que ha sido creado y diseñado por Microsoft para facilitar la programación de software en varias plataformas. A través de Visual Studio, es posible escribir, depurar y probar el código para varios lenguajes, principalmente C# y C++.

Una de las características más importantes de Visual Studio es su capacidad para integrarse con otros servicios y herramientas para que todo el flujo del desarrollo de software, desde la fase inicial hasta el despliegue y la implementación a largo plazo, pueda ser administrado conjuntamente. Adicionalmente, el IDE incorpora herramientas de diseño visual, soporte para entrega y despliegue continuo, y funcionalidades de pruebas automatizadas; lo que lo hace apto para la construcción de una gran variedad de aplicaciones de todo tipo [22]. Para el desarrollo de este componente, se utilizó Visual Studio 2022 community edition.

Git

Git es un sistema de control de versiones distribuido y de código abierto, diseñado para manejar todo tipo de proyectos con velocidad y eficiencia [23]. Este permite la colaboración y seguimiento de los cambios en el código fuente de manera efectiva. También facilita la creación de ramas y fusiones, permitiendo que diferentes funcionalidades sean desarrolladas en paralelo sin interferir unas con otras [23]. Además, su arquitectura distribuida significa que cada desarrollador tiene una copia completa del historial del proyecto, lo que mejora la resiliencia y permite el trabajo fuera de línea.

GitHub

GitHub es una plataforma de desarrollo colaborativo de software que utiliza Git para el control de versiones. GitHub permite a los desarrolladores alojar repositorios de código, colaborar en proyectos y gestionar problemas y solicitudes de extracción (pull requests). Además, ofrece herramientas de integración continua y automatización de flujos de trabajo [24].

Figma

Figma es una herramienta de diseño basada en la nube que permite a los equipos colaborar en tiempo real para crear interfaces de usuario, prototipos y sistemas de diseño. Figma facilita la colaboración en tiempo real, permitiendo el acceso y la edición simultánea por parte de múltiples usuarios. Cuenta con herramientas de edición vectorial, capacidades de prototipado y una amplia biblioteca de elementos de gráficos [25].

SharePoint

SharePoint es una plataforma de colaboración desarrollada por Microsoft que permite a las organizaciones crear sitios web, almacenar, organizar y compartir información de manera segura desde cualquier dispositivo. Gracias a sus características, SharePoint es ideal para mejorar la gestión de documentos, la automatización de flujos de trabajo, así como la comunicación y colaboración dentro de una organización [26].

ClickUp

ClickUp es una herramienta en línea diseñada para que equipos y empresas colaboren y gestionen proyectos de manera eficiente. Ofrece soporte para diversas metodologías y marcos de trabajo, incluyendo Kanban. La plataforma permite asignar tareas, seguir el progreso, crear documentos y gestionar tableros en tiempo real. Además, ClickUp facilita la creación de flujos de trabajo personalizados, el establecimiento de metas y la gestión de recursos mediante una interfaz intuitiva y visual [27].

2.3. Desarrollo

En esta sección, se detalla el proceso para la creación del módulo de administración del sistema. Esta abarca desde la planificación y organización del trabajo hasta el diseño, implementación y pruebas del componente. Se utilizó la metodología ágil Kanban para gestionar el proyecto durante todo el ciclo de vida del desarrollo de software.

La implementación de este módulo tiene como objetivo mejorar la eficiencia de los procesos académicos de los programas de capacitación de LudoLab, así como la administración de la información de los participantes (estudiantes e instructores) y la supervisión del trabajo de los instructores. Para ello, se toma como punto de partida el análisis de los procesos y actividades que se llevan a cabo actualmente en el proyecto de inclusión digital que promueve el laboratorio.

2.3.1. Planificación y análisis

Situación Actual

El proyecto de Inclusión Digital del Laboratorio de Sistemas de Información e Inclusión Digital (LudoLab) de la Escuela Politécnica Nacional, iniciado en 2014, tiene como objetivo democratizar el acceso a las Tecnologías de la Información y las Comunicaciones (TIC). Este proyecto se enfoca en acercar el conocimiento y uso de herramientas tecnológicas esenciales a personas de grupos de atención prioritaria, incluyendo a adultos mayores y jóvenes en situación de vulnerabilidad. Además, busca capacitar a los docentes de escuelas rurales en el uso de estas tecnologías, fundamentales para desenvolverse en el mundo digitalizado en el que vivimos. A través de esta iniciativa, LudoLab pretende dotar a estos grupos de las competencias necesarias para participar activamente en la sociedad digital

[28].

Para ello, y debido al enfoque sin fines de lucro del proyecto, LudoLab hace un llamado al voluntariado a estudiantes universitarios para prestar su servicio y conocimientos en calidad de instructores o de personal administrativo. De este modo, se diferencian los roles que se describen a continuación en la Tabla 2.2:

Tabla 2.2: Roles en el proyecto de inclusión digital

Rol	Descripción
Instructor	Personas voluntarias. Son responsables de impartir los cursos de capacitación en LudoLab. Ellos son los encargados de preparar el contenido del curso, impartir las clases y evaluar el desempeño de los estudiantes.
Estudiante	Personas de grupos de atención prioritarias o docentes de educación básica. Los estudiantes son los beneficiarios de los cursos de capacitación ofrecidos en el proyecto de inclusión digital de LudoLab. Ellos buscan adquirir conocimientos y habilidades a través del programa.
Administrador	Personas voluntarias. Los administradores son responsables de la gestión y supervisión de los participantes de los programas de capacitación en LudoLab. Ellos se encargan de llevar a cabo los procesos académicos y administrativos.

Adicionalmente, la Figura 2.1 muestra un diagrama general con las actividades que se realizan en cada edición del proyecto de inclusión digital de LudoLab.

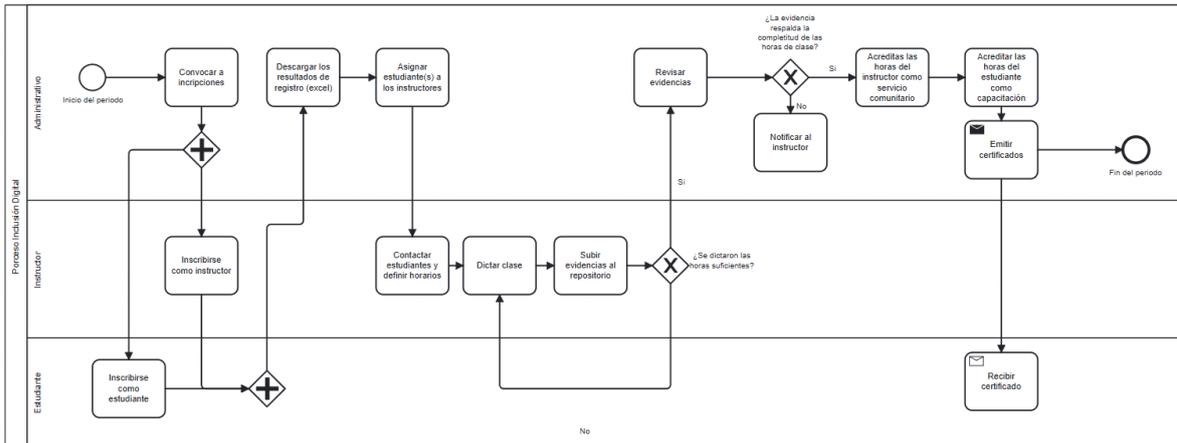


Figura 2.1: Proceso general de Inclusión Digital

Submódulos del componente

En base a las actividades descritas, en adición a una serie de reuniones con los beneficiarios del sistema, se definieron cuatro submódulos que harán parte del componente administrativo, los cuales se detallan a continuación.

- Submódulo de gestión de estudiantes:** Incluye las funcionalidades para crear, editar, eliminar, buscar y listar la información de los estudiantes de los cursos de capacitación de LudoLab.
- Submódulo de gestión de instructores:** Consiste en las funcionalidades para registrar, editar, eliminar, buscar y listar la información de los instructores que harán parte de los cursos de capacitación de LudoLab.
- Submódulo de gestión de cursos:** Permite la creación, configuración y administración de los cursos que se ofrecen en LudoLab, iniciando con el curso de inclusión digital que se oferta en la actualidad.
- Submódulo de subida de archivos:** Permite a los instructores subir archivos de evidencia, como registros de notas, a un repositorio compartido en un sitio de Sharepoint, asegurando al centralización de esta información.

Requerimientos del componente

Después de analizar el proceso general de las actividades académicas y administrativas del programa de inclusión digital, se identificaron 15 requerimientos para el componente administrativo en cuestión. Estos requerimientos, que se distribuyen en los distintos módulos, se muestran en la Tabla 2.3 a manera de historias de usuario, y se pueden encontrar de manera detalla en el Anexo A 4.2:

Tabla 2.3: Historias de usuario para el módulo administrativo

Num	Código	Descripción	Módulo
1	HU01	Como administrador, quiero registrar estudiantes y asignarlos a un curso y período académico para organizar su participación en las clases.	Estudiantes
2	HU02	Como estudiante, quiero ingresar las horas en las cuales estoy disponible para recibir clases, para que el sistema pueda emparejarme con los instructores según disponibilidad.	Estudiantes
3	HU03	Como administrador, quiero ver, crear, leer, actualizar y eliminar los datos de los estudiantes para gestionar su información y participación en los cursos.	Estudiantes
4	HU04	Como administrador, quiero registrar instructores en los cursos para gestionar quién impartirá las clases.	Instructores
5	HU05	Como instructor, quiero ingresar las horas en las cuales estoy disponible para dar clases para que el sistema pueda emparejarme con los estudiantes según disponibilidad.	Instructores
6	HU06	Como administrador, quiero ver, crear, leer, actualizar y eliminar los datos de los instructores para gestionar su información y asignaciones en los cursos.	Instructores

Continúa en la siguiente página

Num	Código	Descripción	Módulo
7	HU07	Como administrador, quiero crear cursos con atributos como nombre y descripción para organizar las materias que se ofrecerán en Ludo-Lab.	Cursos
8	HU08	Como administrador, quiero crear nuevos períodos académicos dentro de cada curso para organizar las diferentes ediciones de los cursos.	Cursos
9	HU09	Como administrador, quiero configurar el número de estudiantes asignados por instructor y el número máximo de grupos que puede tener un instructor para optimizar la organización de las clases.	Cursos
10	HU10	Como administrador, quiero que el sistema empareje automáticamente los horarios de los estudiantes y los instructores para formar grupos de clases según la disponibilidad de ambos.	Cursos
11	HU11	Como administrador, quiero configurar grupos a través de la asignación manual de estudiantes e instructores para mejorar la organización de las clases.	Cursos
12	HU12	Como administrador, quiero ver, crear, leer, actualizar y eliminar la información de los cursos y períodos académicos para gestionar su organización y detalles.	Cursos
13	HU13	Como usuario, quiero acceder a un repositorio de documentos en SharePoint para almacenar los documentos creados como evidencia de trabajo.	Repositorio

Configuración inicial del tablero Kanban

Como se menciona anteriormente en la sección de Metodología, la visualización del progreso del trabajo es una práctica fundamental en la metodología Kanban. Por lo cual, mediante el uso de la herramienta en línea *ClickUp* se ha definido el tablero Kanban de la Figura 2.2 con las fases que se muestran a continuación, de forma que se facilite la visualización del flujo de trabajo.

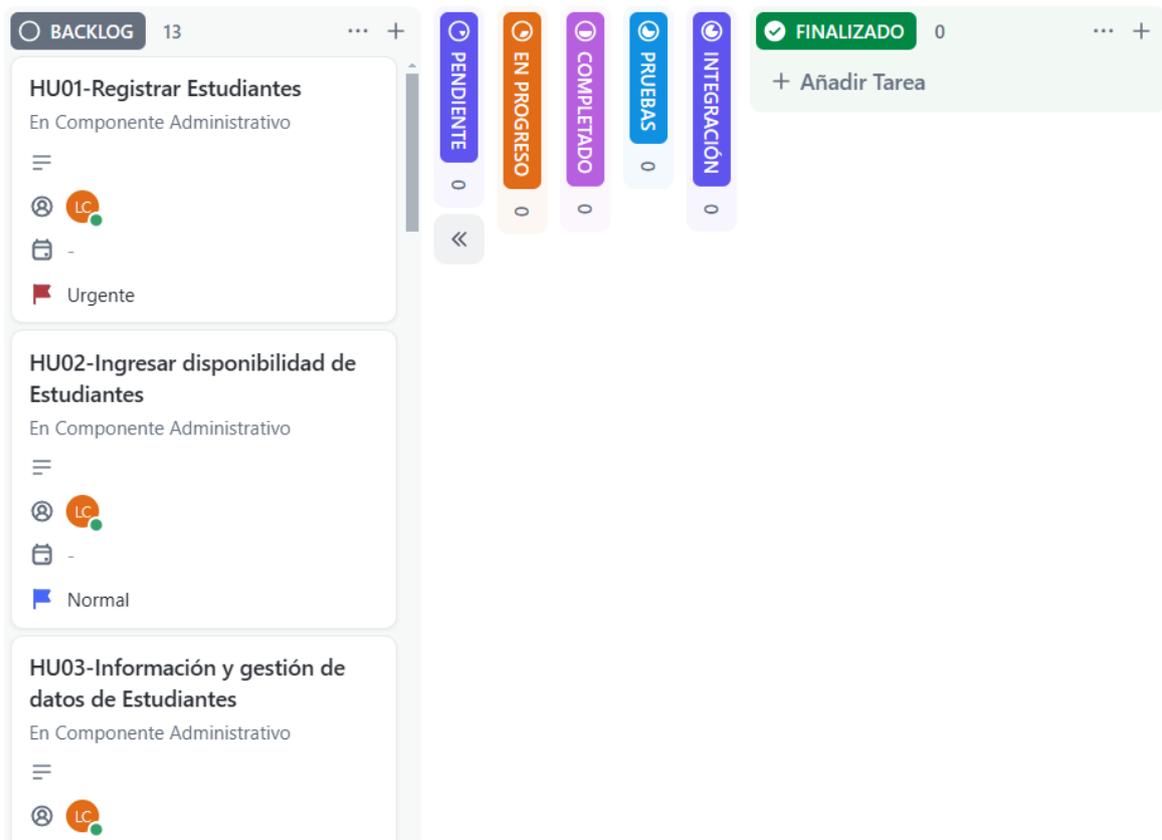


Figura 2.2: Tablero Kanban inicial

De lo cual, se distinguen las siguientes columnas o estados para el tablero:

- Backlog: Es el listado que contiene todas las historias de usuario que aún no han sido iniciadas. Esta columna contiene todos los requerimientos que se priorizarán para el desarrollo en el momento que se considere idóneo.
- Pendiente: Requerimientos que han sido escogidos y priorizados. Están listos para ponerse en marcha.

- En Progreso: Requerimientos que se encuentran en proceso de desarrollo (codificación).
- Completado: Requerimientos que han sido terminados y están listos para su revisión.
- Pruebas: Requerimientos que cuya implementación está siendo probada para asegurar su correcto funcionamiento.
- Integración: Requerimientos que han sido completados y están siendo integrados con el resto del sistema. Para ello, se hace uso de un repositorio grupal en GitHub.
- Finalizado: Requerimientos que han pasado todas las pruebas y se han integrado al sistema sin mayor inconveniente.

Adicionalmente, se acordaron algunas políticas respecto al uso del tablero Kanban en base a los principios y valores de la metodología, que se especificaron en secciones anteriores, estas se muestran a continuación.

- Transparencia en el proceso.- Cada miembro del equipo debe mantener su tablero actualizado, con la finalidad de reflejar el estado real de las tareas que está llevando a cabo.
- Reuniones periódicas.- Si bien se cuentan con tableros individuales para cada miembro del equipo y su componente designado, se realizan reuniones periódicas para discutir acerca del progreso de trabajo, compartir novedades y dar apoyo en los posibles problemas presentados.
- Criterios de finalización.- Una tarea se considera finalizada cuando ha pasado exitosamente por todas las etapas del tablero Kanban.
- Límite del trabajo en progreso (WIP).- Para evitar la sobrecarga de trabajo en los miembros del equipo y en línea con el principio de trabajar con lo que ya está en proceso, se ha establecido un límite máximo de 3 tarjetas por columna en el tablero Kanban.

2.3.2. Diseño

Diagrama de base de datos

La Figura 2.3 muestra el diseño de la base de datos que se utilizó para implementar el componente administrativo.

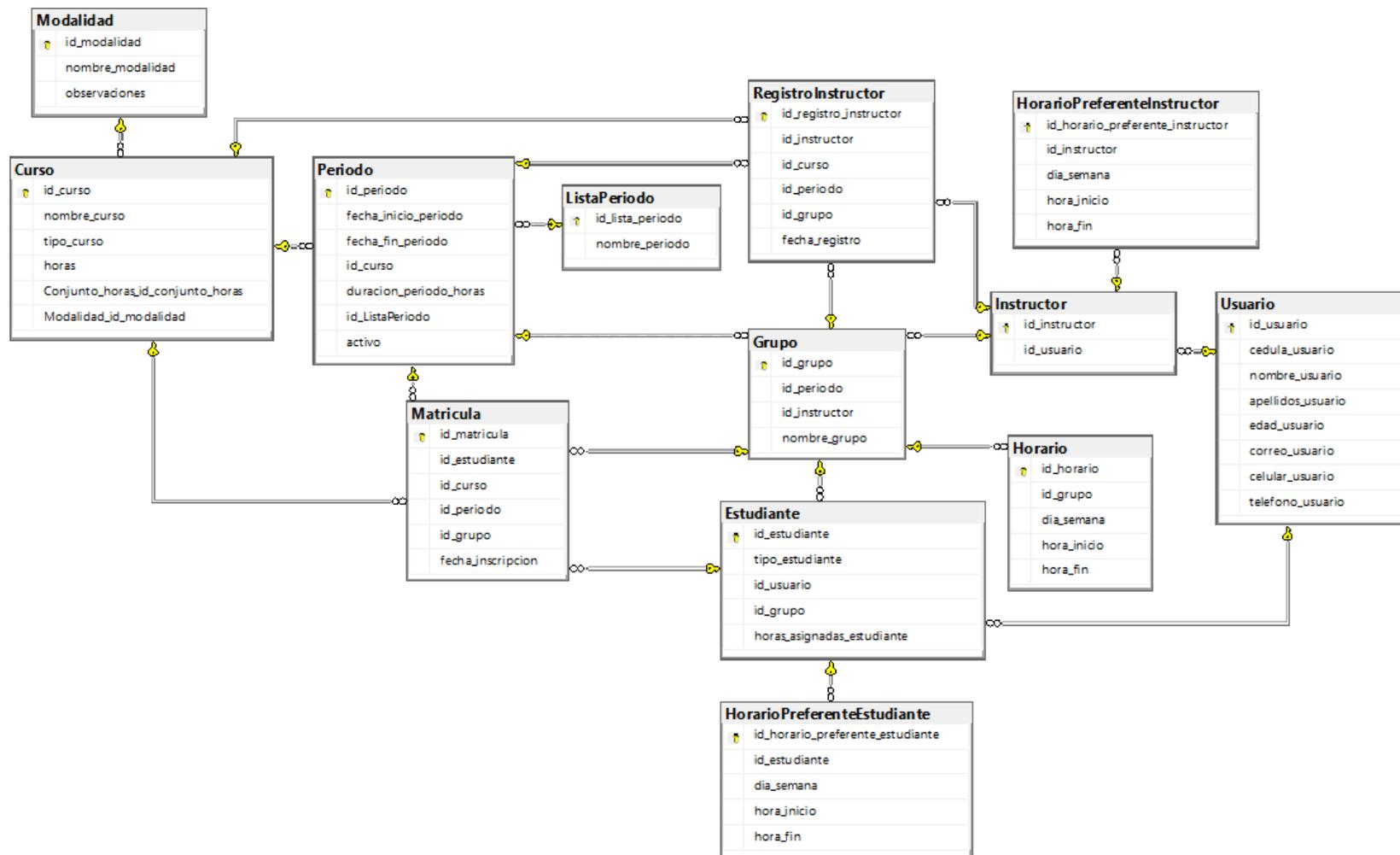


Figura 2.3: Diagrama de base de datos para el componente administrativo desarrollado

Arquitectura

La Figura 2.4 muestra la representación de la arquitectura utilizada para el desarrollo del componente. Esta cuenta con tres capas:

- **Frontend:** El sistema se basa en el uso del framework Blazor WebAssembly, el cual se ejecuta completamente en el lado del cliente. La comunicación entre el cliente y el servidor se realiza mediante peticiones HTTP.
- **Backend:** Para el desarrollo del backend, se utilizó ASP.NET Core Web API. Esta tecnología permitió crear las APIs REST necesarias para proporcionar al cliente los datos solicitados, que se encuentran almacenados en la base de datos.
- **Base de datos:** Se implementó la base de datos representada en SQL Server. En esta capa se albergan los datos que se mostrarán al cliente.

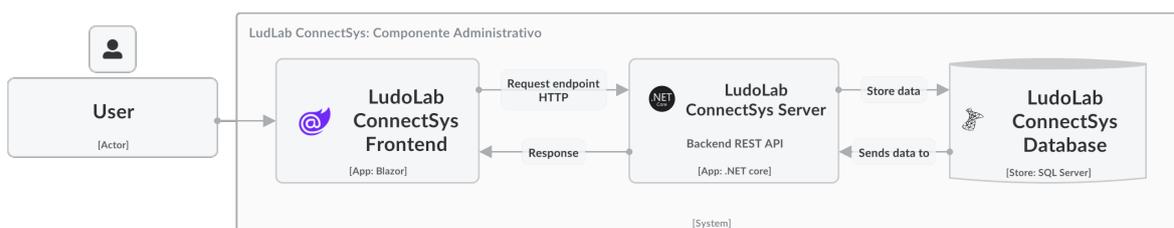


Figura 2.4: Arquitectura de la aplicación

Diseño de interfaces de usuario

Como parte de la fase de diseño del componente, se hizo uso de la herramienta en línea figma para crear los mockups de la aplicación. Esto con la finalidad de tener un acercamiento inicial a la forma y navegabilidad del sistema, obtener retroalimentación de los principales beneficiarios y contar con una guía para llevar a cabo su implementación. A continuación, se muestran algunos mockups diseñados en este primer acercamiento, recalcando que el enlace para acceder a todos los mockups creados se encuentra en el Anexo B 4.2

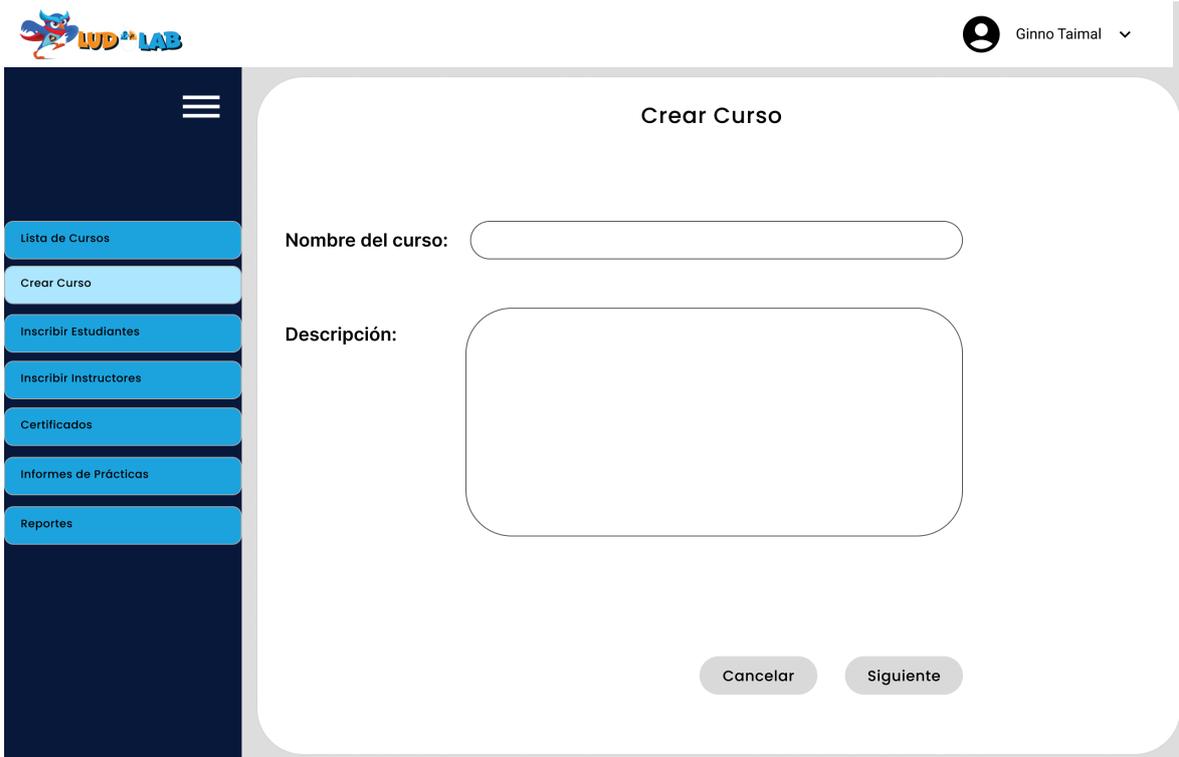


Figura 2.5: Mockup para la interfaz de creación de cursos

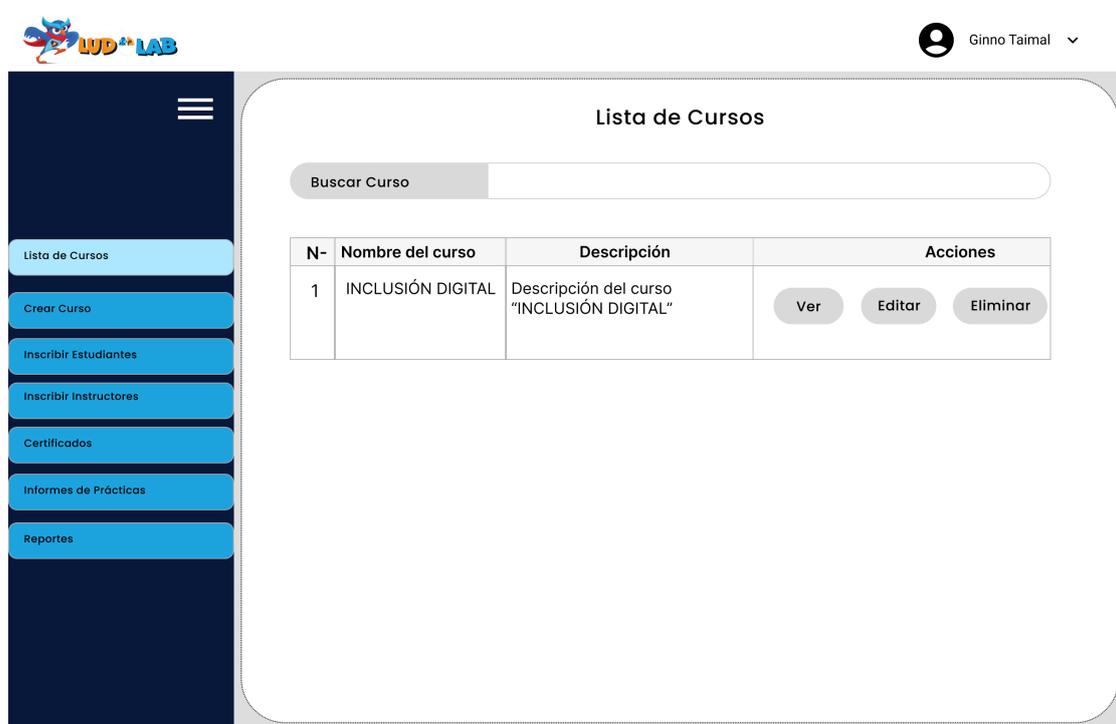


Figura 2.6: Mockup para la interfaz de lista de cursos

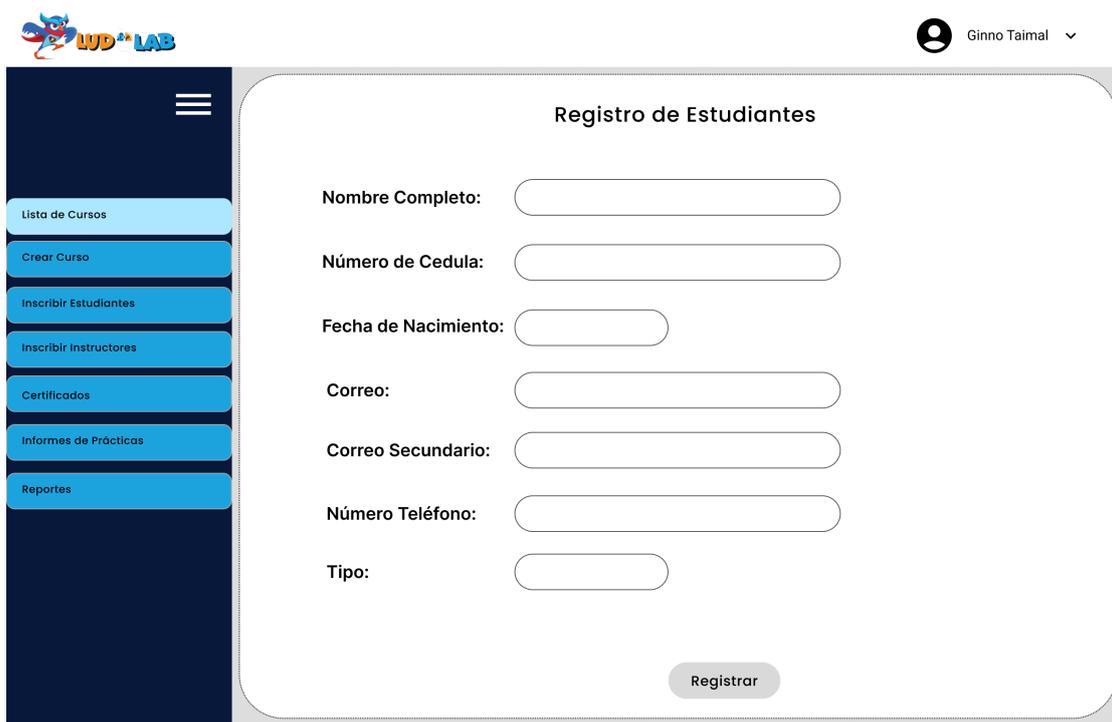


Figura 2.7: Mockup para la interfaz de registro de estudiantes

2.3.3. Implementación

Estructura del proyecto

La estructura del proyecto se ha diseñado a partir de una solución en Visual Studio con múltiples proyectos. Esta se organiza en tres proyectos:

- **Proyecto frontend:**

Desarrollado con Blazor WebAssembly, contiene las interfaces de usuario del componente. Como se mencionó en la sección de herramientas para el desarrollo, Blazor WebAssembly permite construir aplicaciones web interactivas usando C#, de modo que facilita la integración con el backend y mejora la experiencia del usuario con una aplicación de una sola página (SPA). Este proyecto contiene una separación de pantallas por carpetas, las cuales abarcan en su interior las interfaces de usuario para cada rol identificado, como se muestra en la Figura 2.8. Además del directorio /shared, destinado a almacenar componentes compartidos; y wwwroot, que hospeda recursos estáticos como imágenes, archivos de estilo y scripts. A su vez, la Figura 2.9 muestra que cada directorio de área contiene la carpeta /Pages, en cuyo interior existen diferentes carpetas que dividen las interfaces según su funcionalidad; y /Components,

que alberga los componentes utilizados en las interfaces.

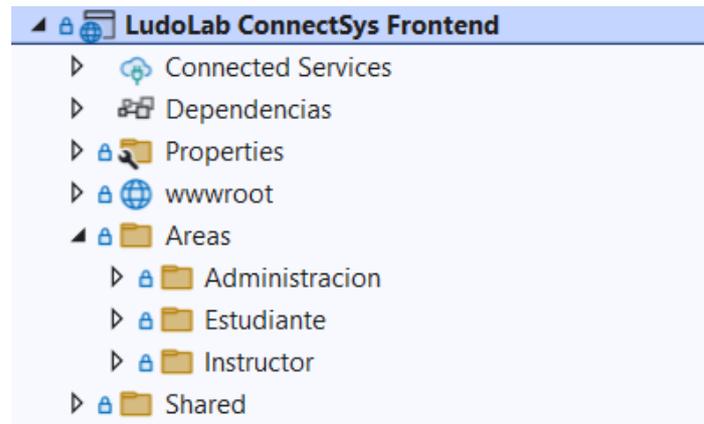


Figura 2.8: Estructura de proyecto frontend general

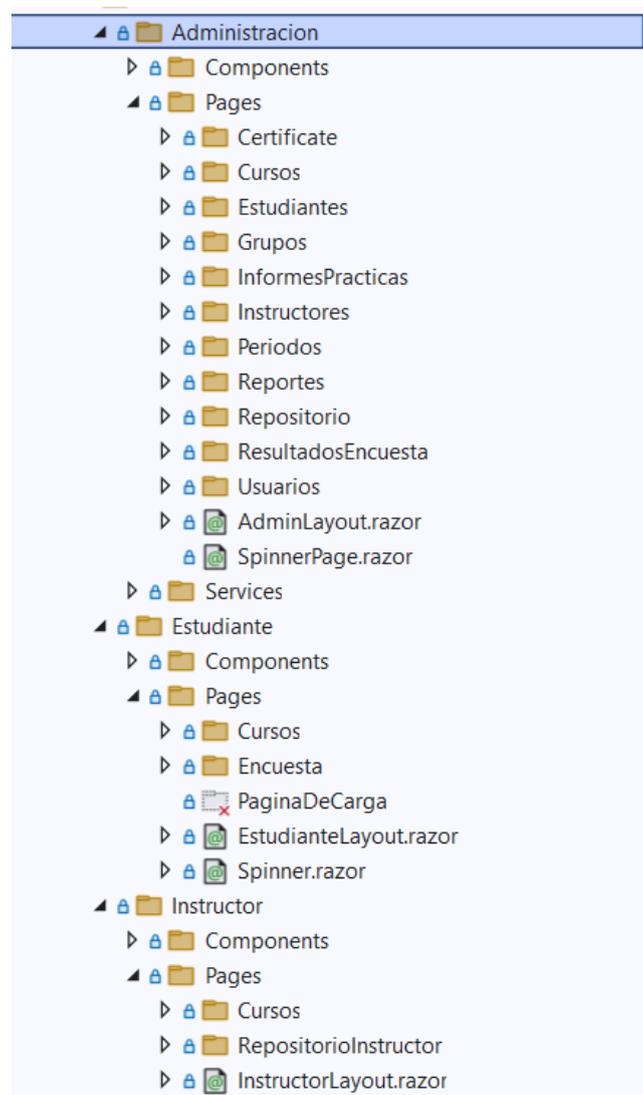


Figura 2.9: Organización de interfaces por áreas (roles)

- **Poyecto backend:**

Desarrollado con ASP.NET Core Web API, proporciona los servicios necesarios para el funcionamiento del componente administrativo. Este proyecto expone una serie de endpoints RESTful que permiten la comunicación con el frontend. Su estructura comprende las carpetas: /Controllers, que maneja las solicitudes HTTP; /Services, que contiene algunos servicios de aplicación; y /Data, que maneja la configuración de acceso a datos y los contextos de base de datos.

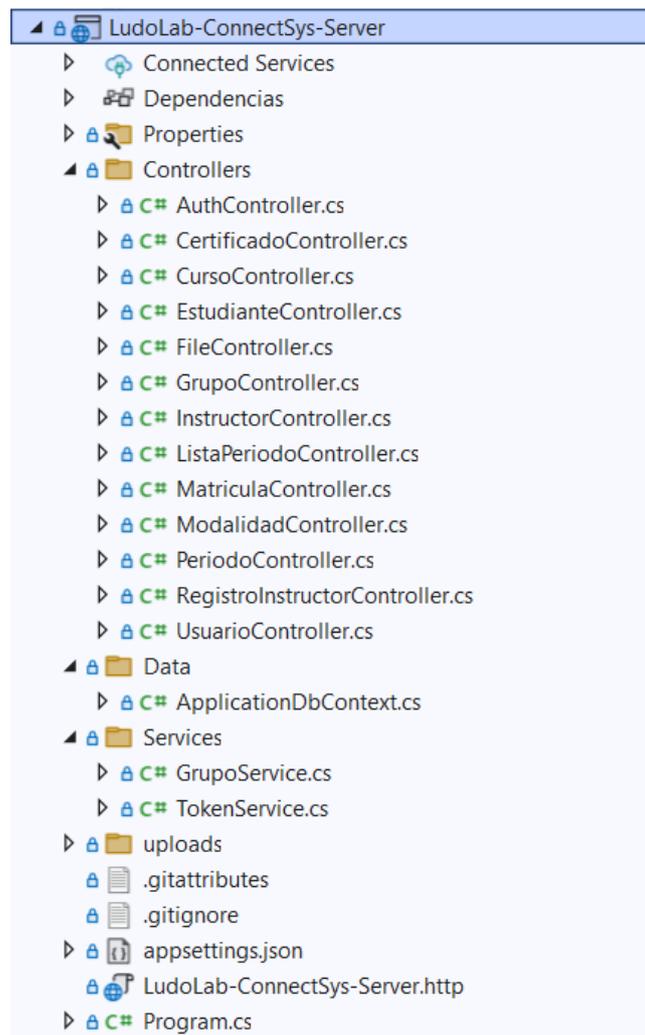


Figura 2.10: Estructura proyecto backend

- **Biblioteca de clases compartida:**

Contiene clases y modelos que son utilizados tanto por el frontend como por el backend. De esta forma, se hace posible contar con definiciones comunes que facilitan una comunicación coherente entre ambos lados. Este enfoque asegura la consistencia en los modelos de datos utilizados en todo el sistema.

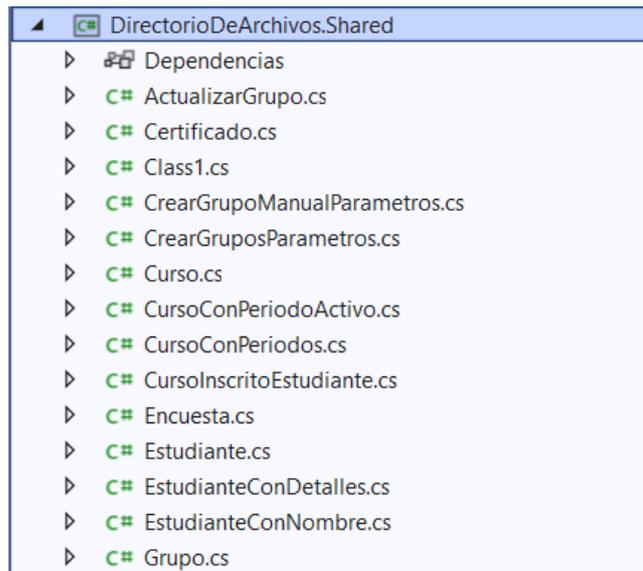


Figura 2.11: Estructura biblioteca de archivos compartidos

Implementación del submódulo Estudiantes

Para iniciar la implementación del componente administrativo, se ha decidido abordar primero las funcionalidades relacionadas con la información estudiantil. Esto se debe a que el módulo de Estudiantes constituye el núcleo del sistema, permitiendo la gestión de estudiantes y sus horarios, lo cual es esencial para el funcionamiento integral del sistema. Además, los módulos de cursos e instructores dependen de la existencia y gestión de estudiantes, haciendo que su implementación sea fundamental para el desarrollo exitoso del resto del proyecto.

Para ello, las historias de usuario que se implementaron fueron:

- **HU01:** Como administrador, quiero registrar estudiantes y asignarlos a un curso y período académico para organizar su participación en las clases.
- **HU02:** Como estudiante, quiero ingresar las horas en las cuales estaba disponible para recibir clases, para que el sistema pudiera emparejarme con los instructores según disponibilidad.
- **HU03:** Como administrador, quiero ver, crear, leer, actualizar y eliminar los datos de los estudiantes para gestionar su información y participación en los cursos.

Estas historias de usuario partieron inicialmente de la fase de **Backlog** del tablero de Kanban definido anteriormente en la sección *Configuración del tablero Kanban*. Sin embargo,

ahora que se ha decidido empezar a trabajar en su implementación, han sido llevadas a la etapa de **Pendiente** dentro del tablero, como se muestra en la Figura 2.12

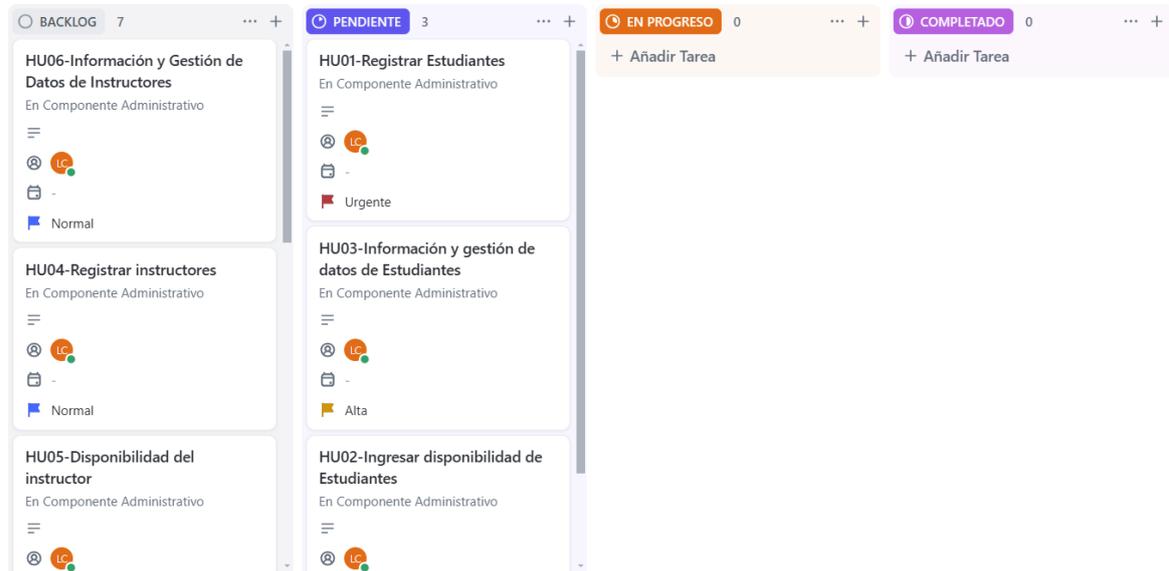


Figura 2.12: Tablero Kanban inicial para el módulo de Estudiantes

A continuación, de las historias de usuario seleccionadas, se eligieron las dos con mayor prioridad para iniciar su proceso de desarrollo. En este caso, y debido a que tienen una relación estrecha, se comenzó a trabajar con las historias HU01 y HU02. Por lo tanto, estas historias se trasladaron a la columna **En Progreso** del tablero Kanban, como se muestra en la Figura 2.13.

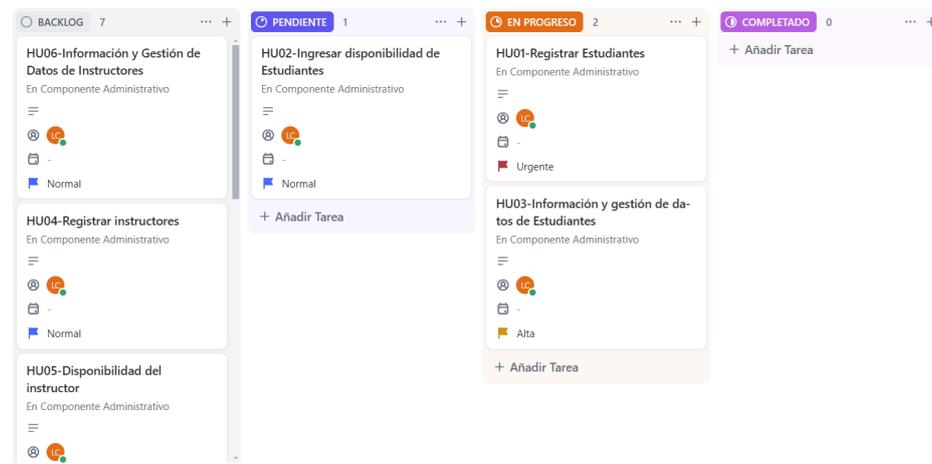
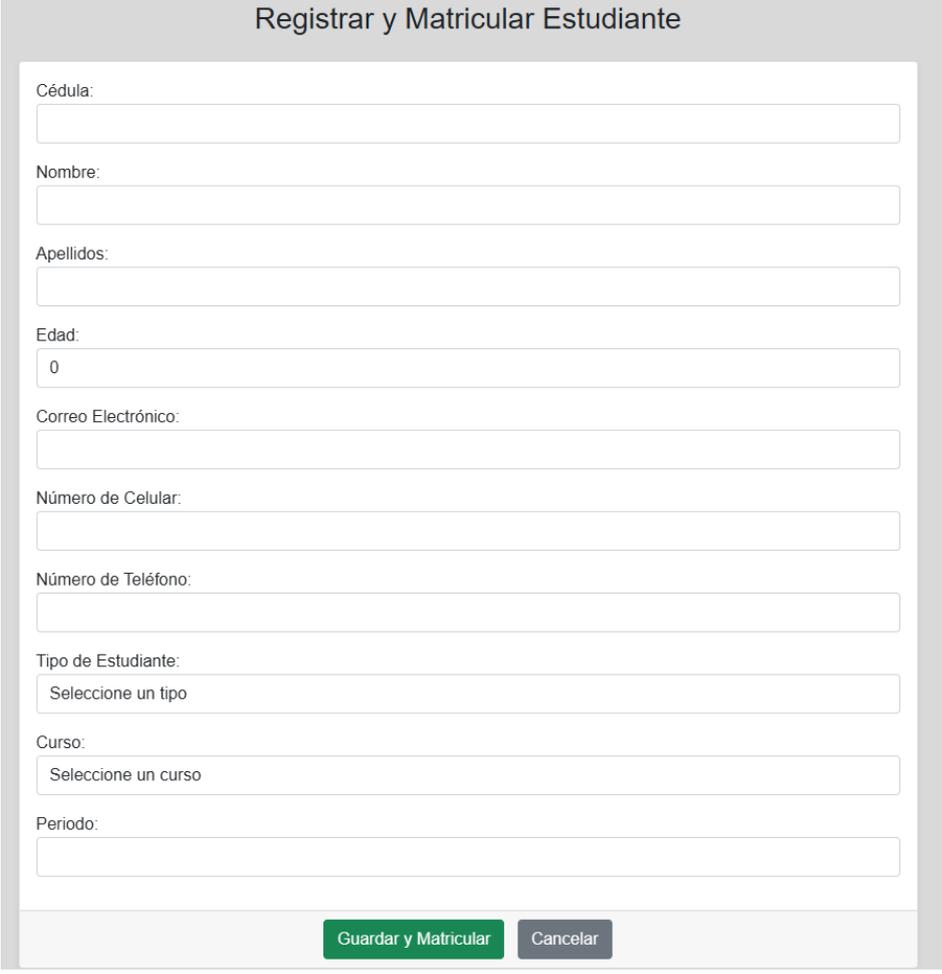


Figura 2.13: Cambio al estado de En Progreso de la HU01 en el tablero Kanban

A continuación, la Figura 2.14 muestra el formulario para el registro y matriculación de

un nuevo estudiante. 2.14.



Registrar y Matricular Estudiante

Cédula:

Nombre:

Apellidos:

Edad:

Correo Electrónico:

Número de Celular:

Número de Teléfono:

Tipo de Estudiante:

Curso:

Periodo:

Figura 2.14: Formulario de registro de estudiante

Por otro lado, la Figura 2.15 muestra la pantalla que contiene la información detallada de un estudiante que ya ha sido escrito de manera previa. Desde esta interfaz no solo es posible visualizar dicha información, si no que también se puede editar la información personal de dicho estudiante, así como eliminar del sistema al estudiante si así lo quisiese el administrador a través del botón de eliminar. Además, se implementó un botón para poder visualizar los cursos en los que el estudiante se encuentra inscrito, cuya funcionalidad se implementará más adelante.

Detalles del Estudiante

Nombre:
David Estudiante

Apellido:
Chiliguano Suarez

Cédula:
1726928532

Edad:
21

Correo:
luis.chiliguano@epn.edu.ec

Celular:
0978665507

Teléfono:
3171333

Tipo de Estudiante:
Grupo de atención prioritaria

[Guardar Cambios](#) [Eliminar](#) [Ver Cursos Inscritos](#)

Figura 2.15: Pantalla de información detallada de un estudiante

Tras implementar las historias de usuario seleccionadas, ahora es posible moverlas hasta el estado de **Completado**. Cabe mencionar que, si bien hace parte del módulo de Estudiantes, la historia de usuario HU02 se implementará más adelante, pues por el momento se han priorizado las funcionalidades presentes en la interfaz que visualiza el Administrador del sistema.

Luego, al cambiar su prioridad de desarrollo, la historia de usuario en cuestión volverá al estado de **Backlog** hasta que se decida cambiar su prioridad. Por otro lado, las historias cuya implementación se ha terminado son movidas al estado de **Completado**, teniendo la disposición que se observa en la Figura 2.16.

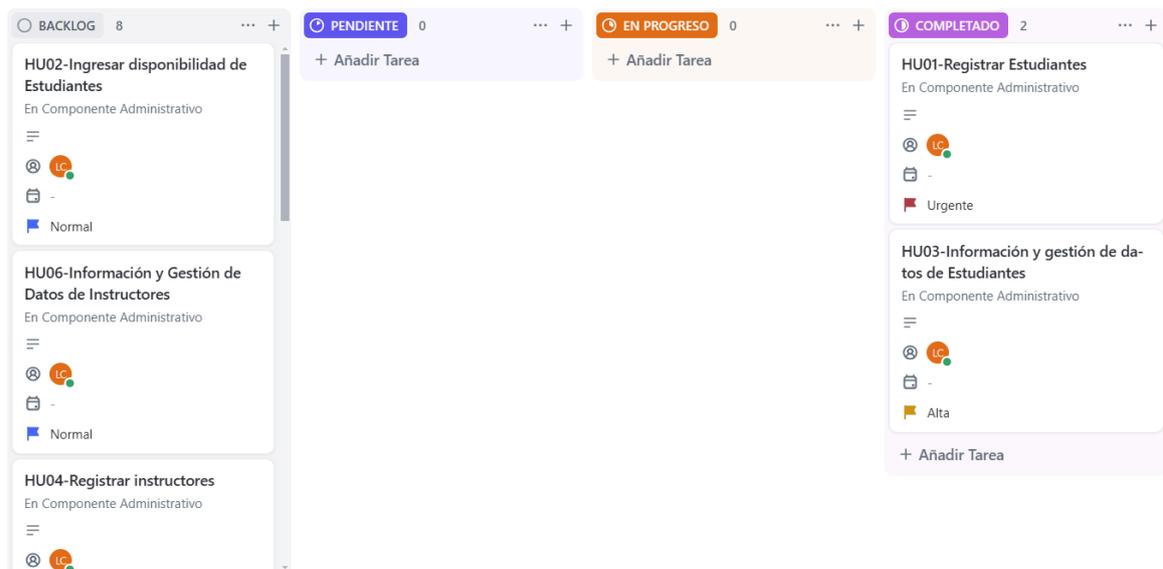


Figura 2.16: Tablero Kanban al completar la programación de las historias de usuario HU01 y HU03

Luego, se iniciaron las pruebas para dichos requerimientos, por lo que pasan de la columna **Completado** a **Pruebas**, como se muestra en la Figura 2.17. La sección de *Pruebas y validación* tiene la información respectiva de los casos de prueba para estas funcionalidades.

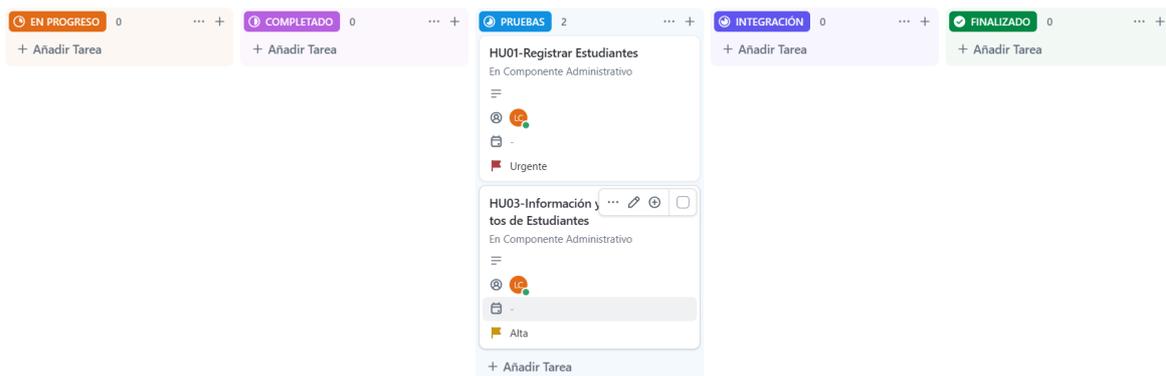


Figura 2.17: Tablero Kanban al iniciar las pruebas para las historias de usuario HU01 y HU03

Una vez completadas las pruebas para estos requerimientos, es posible mover las historias de usuario HU01 y HU03 a la fase de **Integración**, que implica fusionar los cambios

realizados en el proyecto con los realizados en los componentes que se trabajan a la par.

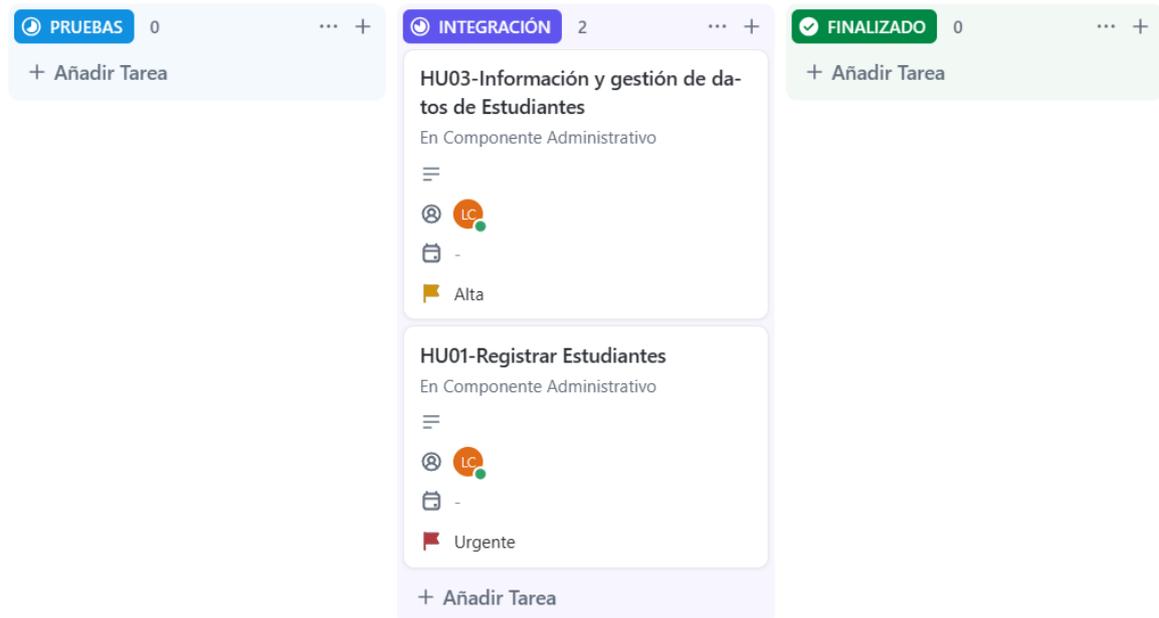


Figura 2.18: Tablero Kanban al iniciar las integración de las funcionalidades descritas en HU01 y HU03

Finalmente, tras realizar la integración de las funcionalidades, es posible mover las historias de usuario HU01 y HU03 al estado de **Finalizado**.

Implementación del submódulo Instructores

Una vez finalizada la implementación del submódulo de información estudiantil, es posible arrastrar al flujo las historias de usuario correspondientes al submódulo de gestión de instructores desde la perspectiva del administrador:

- **HU04:** Como administrador, quiero registrar instructores en los cursos para gestionar quién impartirá las clases.
- **HU06:** Como administrador, quiero ver, crear, leer, actualizar y eliminar los datos de los instructores para gestionar su información y asignaciones en los cursos.

Por lo que, se tiene como tablero inicial al mostrado en la Figura 2.19.

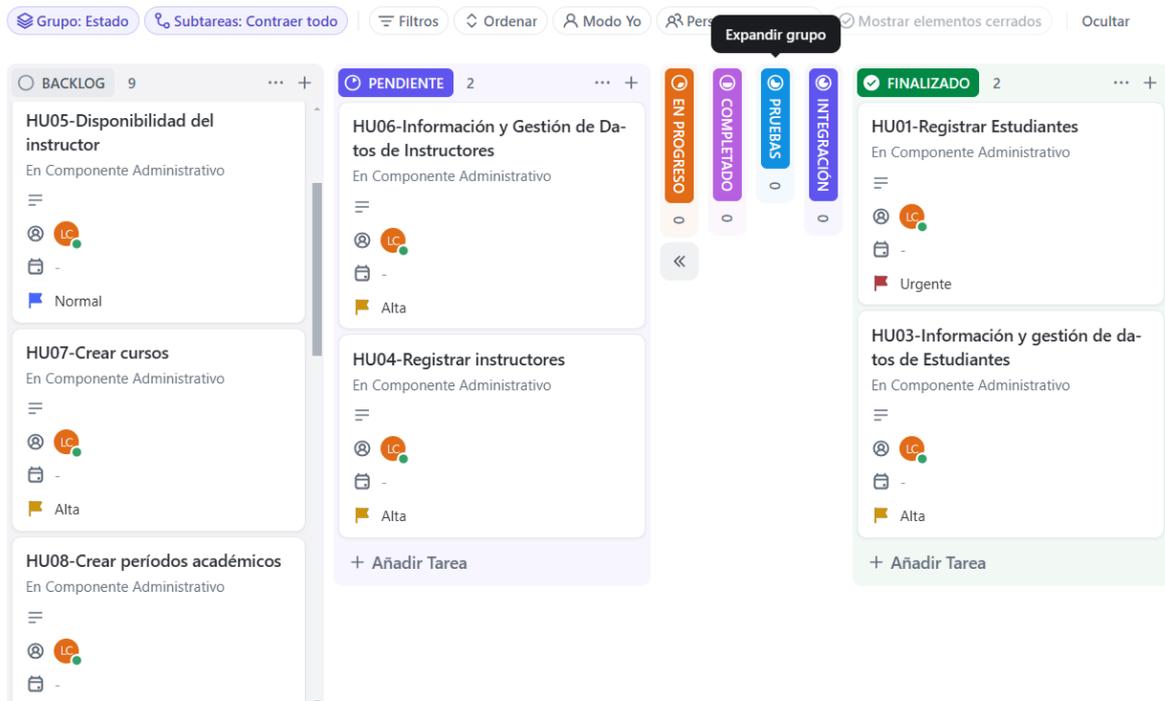


Figura 2.19: Tablero Kanban al iniciar la implementación del submódulo de Instructores

Tras decidir las historias de usuario a desarrollar, es posible moverlas al estado de **En Progreso** dentro del flujo del tablero Kanban. A continuación, la Figura 2.20 muestra el formulario elaborado para registrar un instructor en el sistema.

Registrar Instructor

Cédula:

Nombre:

Apellidos:

Correo Electrónico:

Curso:

Periodo:

Figura 2.20: Formulario de registro de instructores

Adicionalmente, la Figura 2.21 muestra la pantalla que contiene la información detallada de un instructor que ya se ha registrado de forma previa. Además, desde esta interfaz es posible actualizar la información de dicho instructor, así como también existe la posibilidad de eliminarlo del sistema de manera permanente. Adicionalmente, se incluyó un botón para ver los cursos en los cuales el instructor estuvo, o está, registrado para impartir clases, cuya funcionalidad se implementará más adelante.

Detalles del Instructor

Nombre:
Eduardo

Apellido:
Gutierrez

Cédula:
1122334455

Correo:
eduardo.gutierrez@example.com

Celular:
0999888777

Teléfono:
0222112233

Edad:
24

[Guardar Cambios](#) [Eliminar](#) [Ver Cursos Registrados](#)

Figura 2.21: Pantalla de información detallada de un instructor

Luego de haber terminado su codificación, las historias de usuario HU04 y HU06 pueden pasar del estado **Completado** a la fase de **Pruebas**. Los casos de prueba para el módulo de instructores se detallan en la sección de *Pruebas y validación* del presente documento. Posterior a su testeo, es posible integrar las funcionalidades desarrolladas con el resto del proyecto albergado en un repositorio compartido de GitHub. De esta forma, la Figura 2.22 muestra el tablero Kanban final para este módulo.

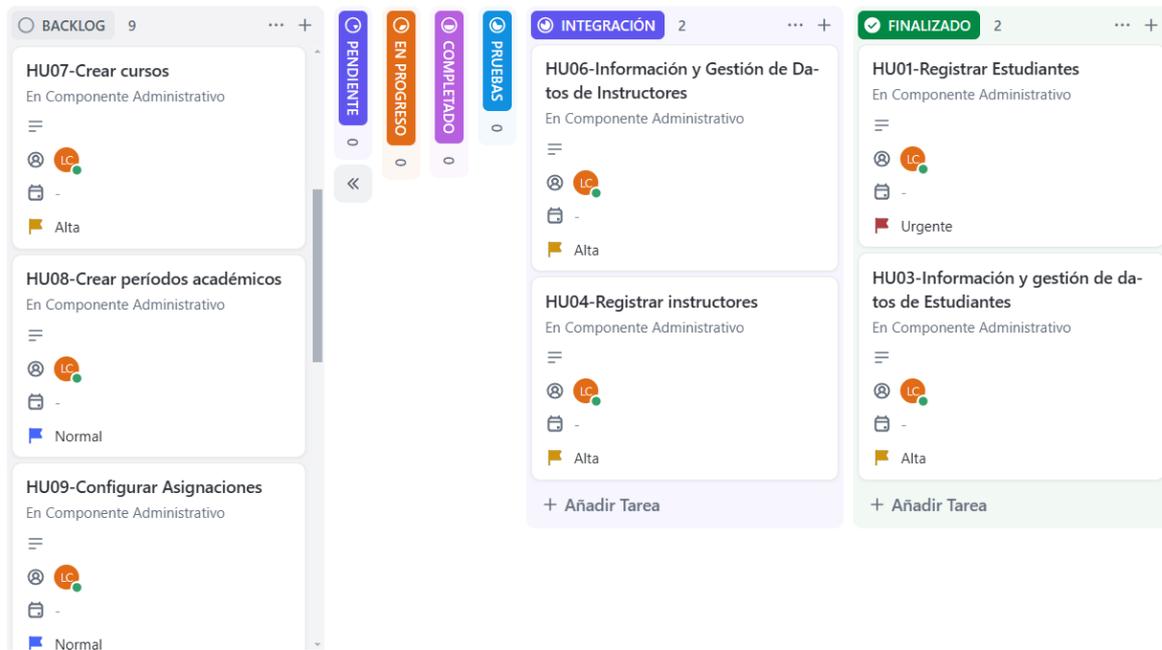


Figura 2.22: Tablero Kanban final para el submódulo de instructores

Implementación del submódulo Cursos

Tras finalizar la implementación del módulo para la gestión de la información de instructores, es posible arrastrar las historias de usuario correspondientes al módulo de cursos dentro del flujo. Para ello, se debe considerar el límite establecido por etapa (columna) del tablero de Kanban, que es de 3 historias de usuario por columna. Por lo cual, se han considerado las siguientes historias de usuario para iniciar la implementación del submódulo en cuestión:

- **HU07:** Como administrador, quiero crear cursos con atributos como nombre para organizar las materias o disciplinas que se ofrecerán en LudoLab.
- **HU08:** Como administrador, quiero crear nuevos períodos académicos dentro de cada curso para organizar las diferentes ediciones de los cursos.
- **HU12:** Como administrador, quiero ver, leer, actualizar y eliminar la información de los cursos y períodos académicos para gestionar su organización y detalles.

Luego, el tablero inicial para la implementación del submódulo de Cursos corresponde a la Figura 2.23.

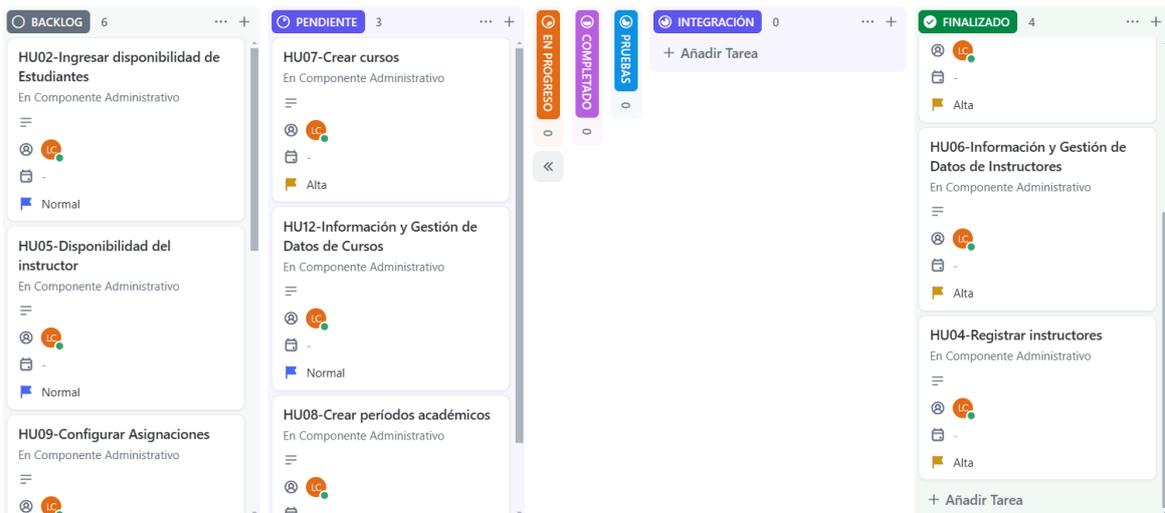


Figura 2.23: Tablero Kanban inicial para el submódulo de Cursos

Posteriormente, se llevaron los requerimientos HU07 y HU08 hacia la columna de **En progreso** para su codificación conjunta, debido a que están estrechamente relacionados. A continuación, la Figura 2.24 muestra el formulario elaborado para agregar un nuevo curso en el sistema, correspondiente al requerimiento HU07.

Crear Nuevo Curso

Nombre del curso:

Tipo del curso:

Horas del curso:

Modalidad:

Crear Curso

Figura 2.24: Formulario para la creación de un nuevo curso

Por otro lado, la Figura 2.25 muestra el formulario elaborado para crear un nuevo periodo de un curso, abordando de esta forma el requerimiento HU08.

Crear Nuevo Periodo

Fecha de Inicio:

Fecha de Fin:

Duración en Horas:

Nombre del Periodo:

[Crear Periodo](#)

Figura 2.25: Formulario para la creación de un nuevo periodo de curso

Una vez que se han implementado las historias de usuario HU07 y HU08, estas fueron llevadas a la etapa de **Completado**. A la par, se decidió arrastrar el requerimiento HU12 a la columna de **En progreso** para empezar su codificación, como se muestra en la Figura 2.26

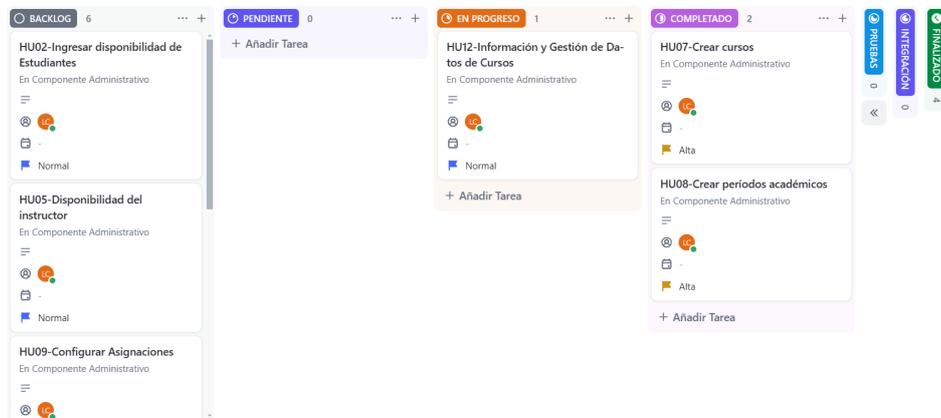


Figura 2.26: Información y gestión de cursos en progreso

La Figura 2.27 muestra el listado de los cursos existentes en el sistema, a partir del cual es posible crear uno nuevo, editar su información, eliminarlo o agregar un nuevo periodo al mismo.

Cursos Disponibles

Crear Nuevo Curso

ID	Nombre	Tipo	Horas	Modalidad	Acciones
1	INCLUSION DIGITAL	Educativo	55	VIRTUAL	Ver Periodos Editar Eliminar
2	Blender	Software 3D	40	PRESENCIAL	Ver Periodos Editar Eliminar
3	JavaScript	Programacion	40	PRESENCIAL	Ver Periodos Editar Eliminar
4	Blazor	Programacion	40	PRESENCIAL	Ver Periodos Editar Eliminar

Figura 2.27: Pantalla de información y gestión de cursos

La funcionalidad de *editar* muestra la interfaz de la Figura 2.28, la cual permite cambiar los datos pre existentes respecto al curso seleccionado.

Editar Curso

Nombre del curso:

Tipo del curso:

Horas del curso:

Modalidad:

Guardar

Figura 2.28: Pantalla de edición de la información general de un curso

Por otro lado, se organizó una lista de periodos para cada curso, como se puede apreciar en la Figura 2.29, de forma que se tiene una interfaz con acciones CRUD para cada elemento de la lista.

Periodos del Curso: INCLUSION DIGITAL

Crear Nuevo Periodo

ID	Fecha de Inicio	Fecha de Fin	Horas	Nombre del Periodo	Acciones
1	25-02-2024	25-02-2025	55	2025A	Editar Eliminar Ver Grupos
12	2024-06-01	2024-06-30	0	2023B	Editar Eliminar Ver Grupos

Figura 2.29: Lista de periodos por curso

Para la funcionalidad de edición de la información de un periodo, se utilizó el formulario que se muestra en la Figura 2.30, muy similar al utilizado para su creación.

Editar Periodo

Fecha de Inicio:

Fecha de Fin:

Duración en Horas:

Nombre de Periodo:

[Guardar](#)

Figura 2.30: Pantalla de edición de un periodo

De esta forma, se ha implementado parcialmente los requerimientos de la HU12. La funcionalidad de *ver grupos* se implementará a medida que sea posible la configuración de los mismos en el sistema en las siguientes historias de usuario. Luego, es posible llevar las historias de usuario HU07, HU08 y HU12 al estado de **Pruebas** dentro del tablero kanban. Los casos de prueba para estos requerimientos, como todos los demás, se especifican más adelante en la sección de *Pruebas y validación* del presente documento. Posteriormente, se procedió con la integración de estas funcionalidades con el repositorio del proyecto en GitHub, a la vez que fue posible arrastrar las siguientes historias

de usuario al flujo de trabajo:

- **HU09:** Como administrador, quiero configurar el número de estudiantes asignados por instructor y el número máximo de grupos que puede tener un instructor para optimizar la organización de las clases.
- **HU10:** Como administrador, quiero que el sistema empareje automáticamente los horarios de los estudiantes y los instructores para formar grupos de clases según la disponibilidad de ambos.
- **HU11:** Como administrador, quiero configurar grupos a través de la asignación manual de estudiantes e instructores para mejorar la organización de las clases.

Es así como se tiene la siguiente disposición para el tablero kanban.

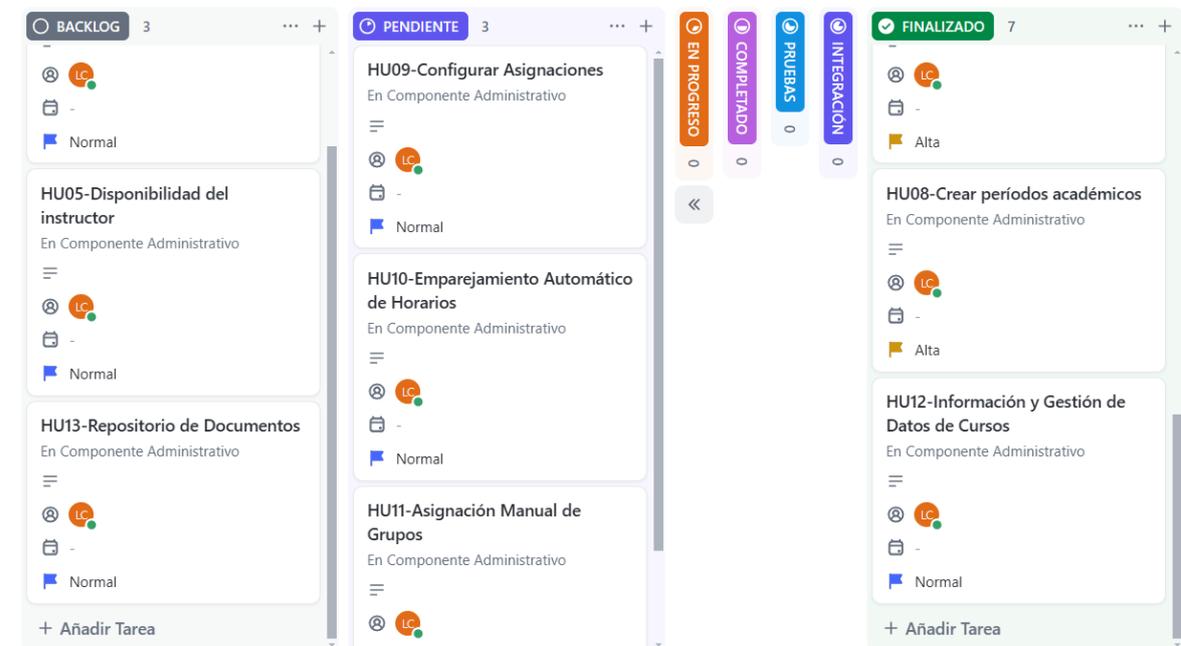


Figura 2.31: Tablero Kanban para implementar requerimientos de grupos

Luego, se empezó a trabajar con los requerimientos HU09 y HU10 de manera simultánea debido a su relación directa. Por lo cual, ambos son llevadas a la columna de **en progreso**. La Figura 2.32 muestra la interfaz utilizada que permite:

1. Establecer los parámetros para la asignación de grupos.
2. Crear grupos de manera automática en base a dichos parámetro de asignación, considerando los horarios de instructores y estudiantes para su emparejamiento.

Crear Grupos

Curso:
INCLUSION DIGITAL

Periodo:
2025A

Número de Estudiantes por Grupo:
1

Número Máximo de Grupos por Instructor:
2

Horas por Clase:
1

[Crear Grupos](#)

Estudiantes sin grupo

ID	Nombre	Horarios Preferentes
8	Roberto Cedeno	Lunes: 08:00:00 - 10:00:00
9	David Basantes	Miercoles: 14:00:00 - 16:00:00

Instructores sin grupo

ID	Nombre	Horarios Preferentes
15	Alvaro	Lunes: 08:00:00 - 10:00:00 Miercoles: 14:00:00 - 16:00:00

Figura 2.32: Parámetros para la creación automática de grupos

Por otro lado, la Figura 2.33 muestra el resultado del emparejamiento de los horarios para proceder con la creación de los grupos provenientes de dicho emparejamiento en la base de datos.

Previsualización de Grupos Creados

Grupo: GR-4
Instructor: Alvaro
Estudiantes:

- Roberto Cedeno

Horario:

- Lunes: 08:00:00 - 10:00:00

Grupo: GR-5
Instructor: Alvaro
Estudiantes:

- David Basantes

Horario:

- Miercoles: 14:00:00 - 16:00:00

[Guardar Grupos](#)

Figura 2.33: Resultados de emparejamiento y creación de grupos

Con respecto al requerimiento HU11, la interfaz para la creación de grupos de forma manual se observa en la Figura 2.34. En esta pantalla, el administrador tiene libertad para escoger al instructor del nuevo grupo, los estudiantes que formarán parte del mismo, y el horario asignado a las clases para dicho grupo.

Crear Grupos Manualmente

Curso:
INCLUSION DIGITAL

Periodo:
2025A

Instructor:
Buscar instructor...
Alvaro Higuain Gustavo

Estudiantes:
Buscar estudiante...
Cristian
Agregar Estudiante

Estudiantes Seleccionados:
 Cristian Eliminar

Horarios:
Lunes
08:00 🕒
10:00 🕒
Eliminar
Agregar Horario
Crear Grupo

Figura 2.34: Asignación manual de estudiantes e instructores para la creación de grupos

Tras terminar su implementación, las historias de usuario HU09, HU10 y HU11 son llevadas desde el estado de **completado** a la fase de **pruebas**. Los casos de prueba para los requerimientos del módulo de cursos se detallan en la sección de *Pruebas y validación* del presente documento. Tras la comprobación de su correcto funcionamiento, fue posible integrar las implementaciones realizadas en el repositorio general del proyecto. La Figura 2.35 muestra el estado final del tablero kanban tras la implementación del módulo de cursos.

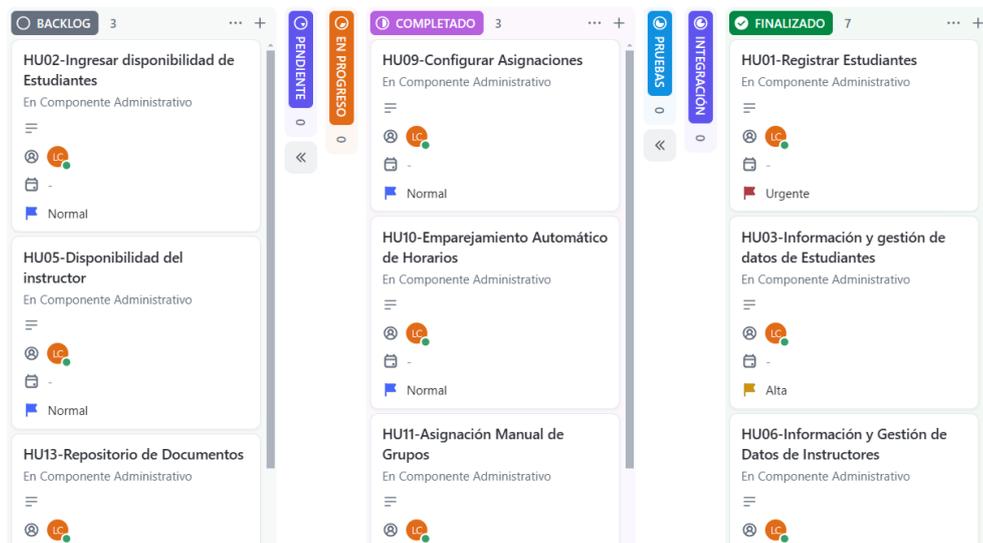


Figura 2.35: Tablero Kanban tras la implementación del submódulo de cursos

Implementación del submódulo de Repositorio

Una vez implementado el módulo para la gestión de cursos, es posible arrastrar la historia de usuario que corresponde al módulo de repositorio, considerando el siguiente requerimiento para su implementación:

- **HU13:** Como instructor, quiero acceder a un repositorio de documentos en SharePoint para almacenar los documentos creados como evidencia de trabajo.

Por lo cual, el tablero inicial para la implementación del submódulo de repositorio es el que se indica en la Figura 2.36.

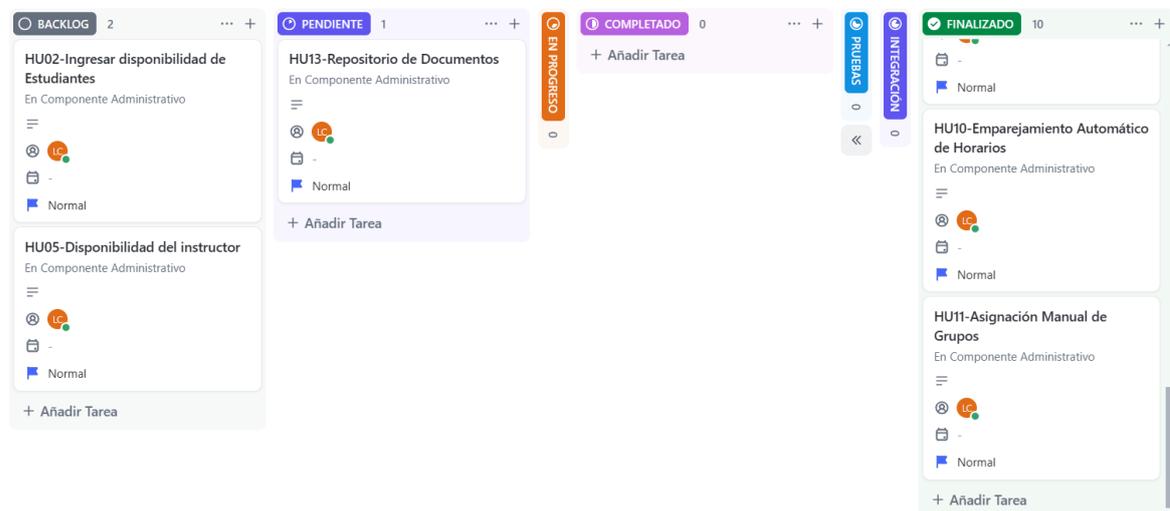


Figura 2.36: Tablero kanban inicial para el submódulo de repositorio

Posteriormente, se llevó el requerimiento HU13 hacia la columna de **en progreso** para su codificación.

Para el desarrollo de esta funcionalidad, fue necesario utilizar Microsoft Graph, una plataforma que conecta múltiples servicios y dispositivos de Microsoft a través de una API única [29]. Esto permitió acceder a los sitios de SharePoint disponibles en el dominio donde está registrada la aplicación desarrollada. Para que esta funcionalidad opere correctamente, es esencial contar con un inquilino en Azure de Microsoft, lo que permite el registro de aplicaciones y la gestión de permisos necesarios para acceder y manejar los recursos de SharePoint de manera segura.

Luego, fue necesario importar, entre otras, la dependencia que permite la conexión con Microsoft Graph (Figura 2.37).

```

@page "/repositorio"
@using Microsoft.AspNetCore.Components.Authorization
@using Microsoft.AspNetCore.Components.WebAssembly.Authentication
@using Microsoft.Graph
@using System.Net.Http.Headers
@using System.Text.Json
@inject GraphServiceClient GraphClient
@inject NavigationManager Navigation
@inject HttpClient Http

```

Figura 2.37: Dependencias necesarias para la interfaz de repositorio

Entonces, se muestra una interfaz de usuario que depende del estado de autenticación del usuario, teniendo que:

- No Autenticado: Si el usuario no está autenticado con una cuenta del dominio en el cual se ha registrado la aplicación, se le muestra un mensaje indicando que debe iniciar sesión para subir archivos, junto con un enlace para autenticarse.
- Autenticado: Si el usuario está autenticado, se muestra un control para seleccionar archivos (*InputFile*) y un botón para subir el archivo seleccionado.

La lógica del código para la autenticación se muestra en la Figura 2.38

```

<AuthorizeView>
  <NotAuthorized>
    <div>Debe autenticarse para subir archivos</div>
    <p>Haga clic en <a href="authentication/login">Iniciar sesión</a>
      para autenticarse y subir archivos al sitio de SharePoint.</p>
  </NotAuthorized>
  <Authorized>
    <InputFile OnChange="HandleSelected" accept=".pdf,.docx,.xlsx,.jpg,.png,.txt" />
    <button @onclick="UploadFile">Subir Archivo</button>
    @if (uploadResult != null){...}

```

Figura 2.38: Lógica de autenticación

A continuación, la Figura 2.39 muestra la interfaz que pedirá la autenticación del usuario en el dominio de Azure en el cual se encuentre registrada la aplicación. Sin autenticarse de esta manera, será imposible cargar archivos al sitio designado.

Subir Archivos a Sharepoint

Debe autenticarse para subir archivos

Haga clic en [Iniciar sesión](#) para autenticarse y subir archivos al sitio de SharePoint.

Figura 2.39: Interfaz inicial de repositorio

Tras autenticarse, el usuario podrá visualizar la interfaz de la Figura 2.40, la cual le permite seleccionar un archivo de su almacenamiento local para proceder a subirlo al repositorio de Sharepoint.

Subir Archivos a Sharepoint

Elegir archivo

No se ha seleccionado ningún archivo

Subir Archivo

Figura 2.40: Interfaz para subir archivo

En cuanto al código (Figura 2.41), se maneja el evento de selección de archivo y almacena el archivo seleccionado en *selectedFile*.

```

<Authorized>
  <InputFile OnChange="HandleSelected" accept=".pdf,.docx,.xlsx,.jpg,.png, .txt" />
  <button @onclick="UploadFile">Subir Archivo</button>
  @if (uploadResult != null)
  {
    <p>Resultado de la carga: @uploadResult</p>
  }
  @code {

private IBrowserFile? selectedFile;
private string? uploadResult;

// Valores crudos (obtén estos valores de tu portal de SharePoint)

private readonly string siteId = "ID del Sitio de Sharepoint";
private readonly string driveId = "ID de la carpeta destinada dentro del Sitio";

private void HandleSelected(InputFileChangeEventArgs e)
{
  selectedFile = e.File;
}
}

```

Figura 2.41: Lógica para la selección de archivos

De este modo, será posible examinar el explorador de archivos para seleccionar el elemento que se desea subir.

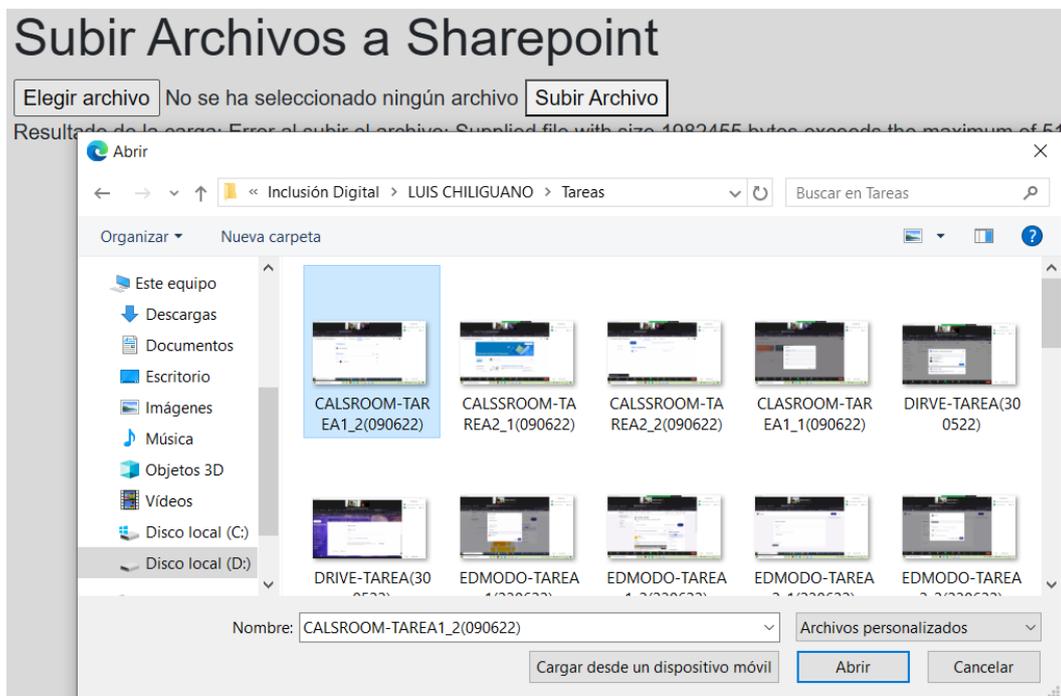


Figura 2.42: Selección de elemento a subir

Cuando el usuario hace clic en el botón "Subir Archivo", se ejecuta el método *UploadFile* (Figura2.43) que realiza las siguientes acciones:

- Obtiene un token de acceso llamando al método *GetAccessToken*, a través de una solicitud HTTP.
- Verifica que se haya seleccionado un archivo.
- Abre el archivo como un flujo de datos y crea una solicitud HTTP PUT a Microsoft Graph API para subir el archivo a una carpeta específica en el sitio de SharePoint, utilizando el *driveId* y *siteId* configurados.
- Si la carga es exitosa, se muestra un mensaje de éxito; en caso de error, se muestra un mensaje con los detalles del error.

```
private async Task UploadFile()
{
    Console.WriteLine("INGRESO AL UploadFile");
    if (selectedFile != null)
    {
        Console.WriteLine($"Archivo seleccionado: {selectedFile.Name}");
        try
        {
            var tokenResult = await GetAccessToken();
            using (var stream = selectedFile.OpenReadStream())
            {
                Console.WriteLine("INGRESO AL OpenReadStream");
                var requestUrl = "https://graph.microsoft.com/v1.0/Drives/" + driveId + "/root:/ejemplo/"
                    + selectedFile.Name
                    + ":/content";
                Console.WriteLine($"REQUEST URL: {requestUrl}");

                var request = new HttpRequestMessage(HttpMethod.Put, requestUrl)
                {
                    Content = new StreamContent(stream)
                };

                var contentType = string.IsNullOrEmpty(selectedFile.ContentType)
                    ? "application/octet-stream"
                    : selectedFile.ContentType;
                request.Content.Headers.ContentType = new MediaTypeHeaderValue(contentType);
                request.Headers.Authorization = new AuthenticationHeaderValue("Bearer", tokenResult);

                var response = await Http.SendAsync(request);
                response.EnsureSuccessStatusCode();
                uploadResult = "Archivo subido exitosamente.";
            }
        }
    }
}
```

Figura 2.43: Método para subir archivos a Sharepoint

Finalmente, tras accionar el botón "Subir Archivo" se realizará la carga respectiva al sitio de SharePoint que se haya configurado de forma previa.

Subir Archivos a Sharepoint

Elegir archivo

CALSROOM-TAREA1_2(090622).png

Subir Archivo

Resultado de la carga: Archivo subido exitosamente.

Figura 2.44: Resultado de carga exitosa

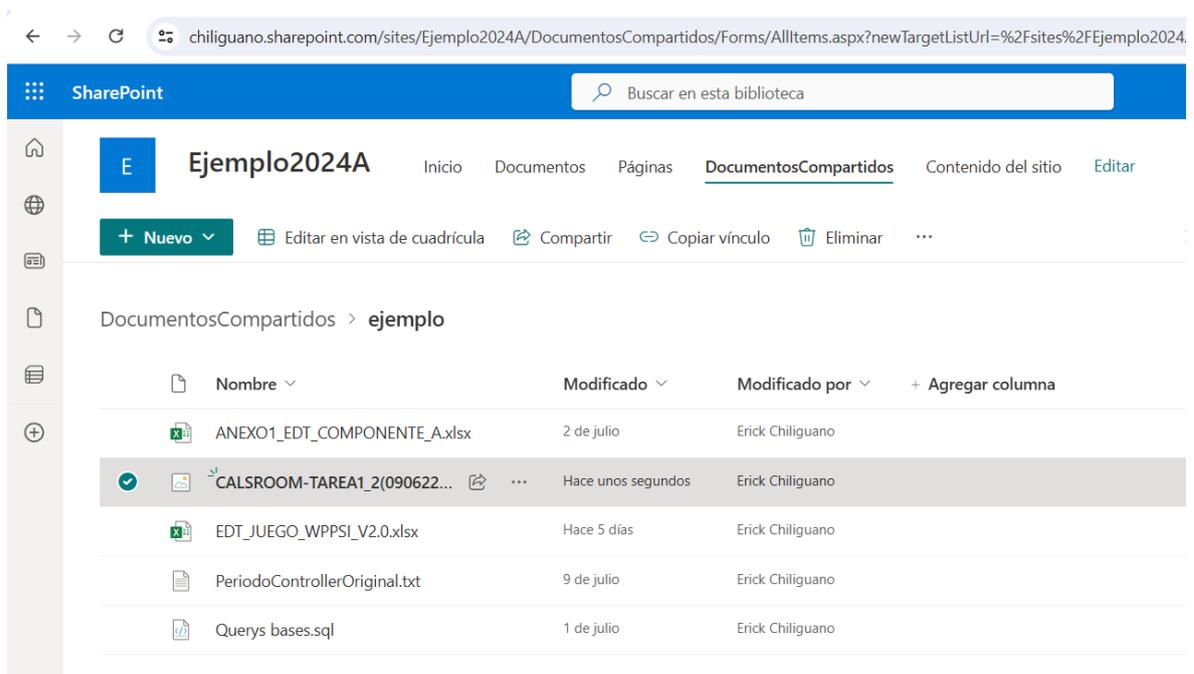


Figura 2.45: Carga de archivo exitoso en el repositorio de Sharepoint

Tras terminar su implementación, la historia de usuario HU13 es llevada desde el estado de **completado** a la fase de **pruebas**. Los casos de prueba para el requerimiento del módulo de repositorio se detallan en la sección de *Pruebas y validación*. Tras la comprobación de su correcto funcionamiento, fue posible integrar la implementación realizada en la rama de integración del repositorio del proyecto. La Figura 2.46 muestra el estado final del tablero

kanban tras la implementación del módulo de repositorio de documentos.

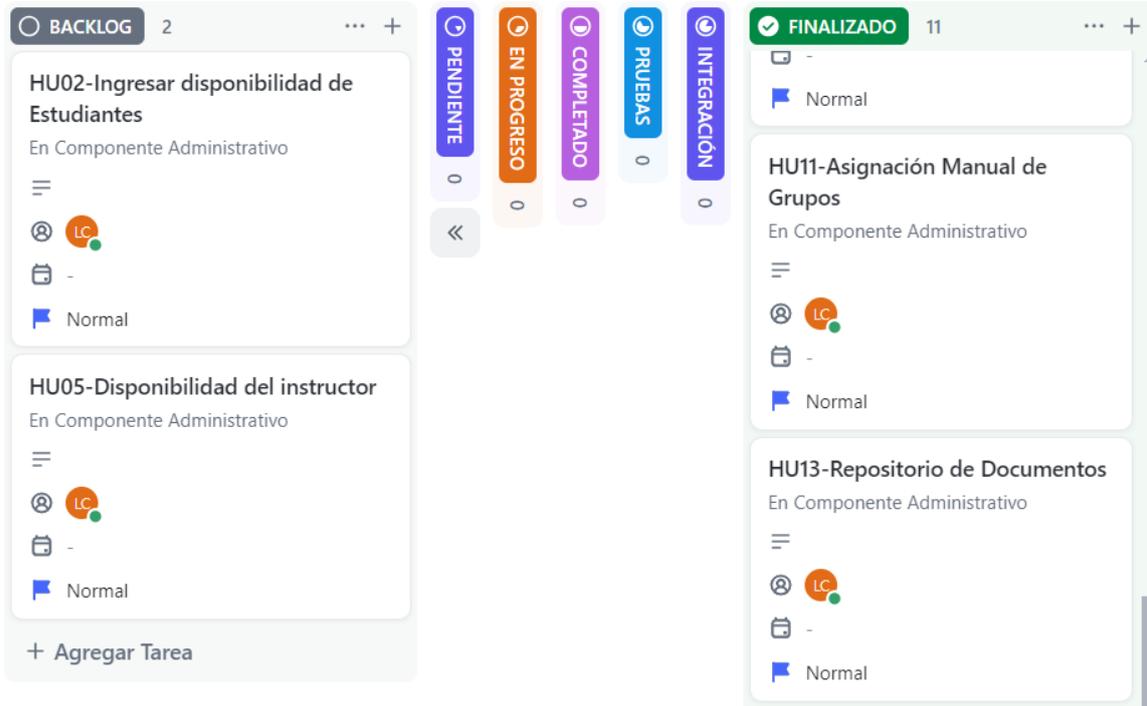


Figura 2.46: Tablero Kanban tras la implementación del submódulo de repositorio de documentos

Para finalizar, se implementarán las historias de usuario que quedaron pendientes en el apartado de los módulos de estudiantes e instructores. En primer lugar, se tiene la historia de usuario:

- **HU02:** Como estudiante quiero ingresar las horas en las cuales estoy disponible para recibir clases para que el sistema pueda emparejarme con los instructores según disponibilidad.

Para abordarla, se utilizó la interfaz de la Figura 2.47 para mostrar los cursos disponibles para que el estudiante realice su inscripción.

Cursos Disponibles

INCLUSION DIGITAL	Inscribirse
Blender	Inscribirse
JavaScript	Inscribirse

Figura 2.47: Cursos disponibles para inscripción

Al seleccionar uno de estos, el usuario estudiante puede ingresar sus horarios preferentes, como se aprecia en la Figura 2.48.

Ingresar Horarios Preferentes

Día de la Semana:

Hora de Inicio:

Hora de Fin:

Horarios Preferentes

- Martes: 08:00:00 - 10:00:00

Figura 2.48: Ingreso disponibilidad estudiante

De forma análoga, para abordar la historia de usuario HU05

- **HU05:** Como instructor quiero ingresar las horas en las cuales estoy disponible para dar clases para que el sistema pueda emparejarme con los estudiantes según disponibilidad.

Se utiliza la interfaz de la Figura 2.49, en la cual el usuario es capaz de escoger el curso al cual quiere registrarse como instructor:

Cursos Disponibles

INCLUSION DIGITAL
Periodo: 2025A
Inicio: 25-02-2024 - Fin: 25-02-2025
[Registrarse](#)

Blender
Periodo: 2025A
Inicio: 03-06-2022 - Fin: 03-08-2022
[Registrarse](#)

JavaScript
Periodo: 2024A
Inicio: 2024-06-01 - Fin: 2024-08-31
[Registrarse](#)

Figura 2.49: Cursos disponibles para registro como instructor

Además, el ingreso de disponibilidad para el instructor se realiza a través de un formulario similar al utilizado para estudiantes, como se muestra en la Figura 2.50

Ingresar Horarios Preferentes

Día de la Semana:
Seleccione un día

Hora de Inicio:
00:00

Hora de Fin:
00:00

[Agregar Horario](#)

Horarios Preferentes

- Martes: 10:00:00 - 11:00:00 [Eliminar](#)
- Viernes: 14:00:00 - 16:00:00 [Eliminar](#)

[Guardar Horarios](#)

Figura 2.50: Ingreso de disponibilidad de un instructor

Tras su implementación, prueba e integración, las historias de usuario HU02 y HU05 son llevadas al estado de **finalizado** en el tablero de Kanban, teniendo como tablero final y resultante tras la implementación de todos los requerimientos especificados el que se observa en la Figura 2.51.

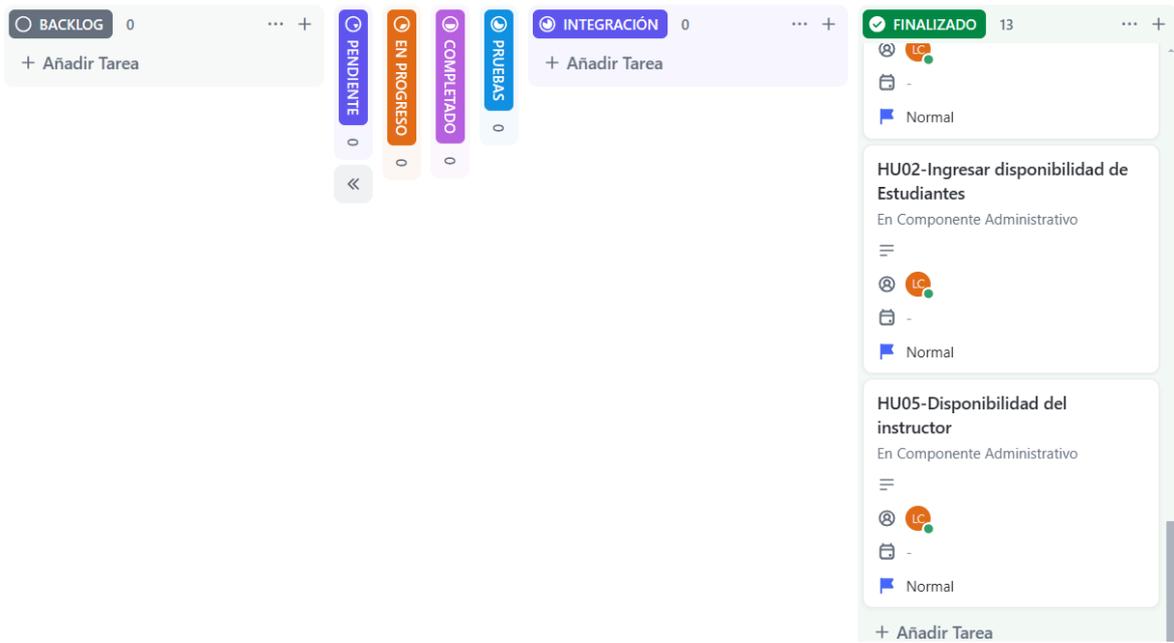


Figura 2.51: Tablero Kanban tras finalizar la implementación del componente administrativo en su totalidad

Capítulo 3

EVALUACIÓN Y RESULTADOS

Las pruebas en el desarrollo de software consisten en evaluar y comprobar que un producto o aplicación de software funciona según lo previsto [30]. Por lo cual, representan una etapa crítica en el ciclo de vida de su desarrollo, pues permiten identificar y corregir defectos antes de que el producto sea entregado a los usuarios finales, lo que contribuye a mejorar la calidad y fiabilidad del mismo.

En este capítulo, se exponen los resultados de las pruebas de funcionalidad, realizadas mediante casos de prueba, y las pruebas de usabilidad aplicadas al componente desarrollado.

3.1. Pruebas de funcionalidad

Como se mencionó anteriormente, para realizar las pruebas de funcionalidad del componente se emplearon casos de prueba diseñados específicamente para validar el correcto funcionamiento de cada elemento del mismo. Estos casos de prueba se enfocaron en verificar las funcionalidades clave de cada submódulo del componente, tomando en cuenta los requisitos especificados para cada uno de estos. Se consideraron también las condiciones iniciales, el procedimiento a seguir, el resultado esperado y el resultado obtenido. Finalmente, para mantener un control sobre los casos aprobados, se incorporó un campo que indica si el caso fue aprobado o no. A continuación, se presentan algunos de los casos de prueba implementados, recalando que el documento que contiene todos los casos de prueba se encuentra en el Anexo C 4.2.

La Figura 3.1 muestra el caso de prueba utilizado para comprobar la funcionalidad del requerimiento *HU01-Registrar estudiantes*.

Caso de prueba: Registro exitoso de estudiantes	
Numero: 1	Usuario: Administrador
Historia de usuario relacionada: HU01-Registrar estudiantes	
Condiciones iniciales:	Acceso al sistema como administrador
Procedimiento:	<ol style="list-style-type: none"> 1. Navegar a la interfaz de registro de estudiantes 2. Ingresar la información del estudiante a registrar: cédula, nombre, apellidos, edad, correo electrónico, números de contacto. 3. Escoger el tipo de estudiante 4. Seleccionar un curso y un periodo académico 5. Hacer click en el botón de "guardar y matricular".
Resultado esperado:	El sistema muestra un mensaje de confirmación y el estudiante aparece como seleccionable para la creación de grupos.
Resultado obtenido:	El resultado es el esperado
Cumple:	Si

Figura 3.1: Caso de prueba para el registro exitoso de estudiantes

La Figura 3.2 muestra el éxito del caso de prueba *Registro Exitoso de Estudiantes*.

The image shows a web form for student registration. The form fields are filled with the following information:

- Nombre: Roberto
- Apellidos: Crespo Suarez
- Edad: 21
- Período: 2025A

A modal dialog box is displayed in the center of the screen, indicating a successful registration. The dialog contains a green checkmark icon and the text: "Estudiante creado y matriculado exitosamente". Below the text is a blue "OK" button. At the bottom of the form, there are two buttons: "Guardar y Matricular" (highlighted in green) and "Cancelar".

Figura 3.2: Registro exitoso de un estudiante

La Figura 3.3 muestra el caso de prueba utilizado para comprobar la funcionalidad del requerimiento *HU04-Registrar instructores*.

Caso de prueba: Registro exitoso de instructores	
Numero: 6	Usuario: Administrador
Historia de usuario relacionada: HU04-Registrar instructores	
Condiciones iniciales:	Acceso al sistema como administrador
Procedimiento:	<ol style="list-style-type: none"> 1. Navegar a la interfaz de registro de instructores 2. Ingresar la información del instructor a registrar: cédula, nombre, apellidos, edad, correo electrónico, números de contacto 3. Seleccionar un curso y un período académico 4. Hacer click en el botón de "guardar y registrar".
Resultado esperado:	El sistema muestra un mensaje de confirmación y el instructor aparece en la lista de instructores
Resultado obtenido:	El resultado es el esperado
Cumple:	Si

Figura 3.3: Caso de prueba para el registro exitoso de instructores

De manera análoga al caso anterior, la Figura 3.4 muestra el éxito del caso de prueba *Registro Exitoso de Instructores*.

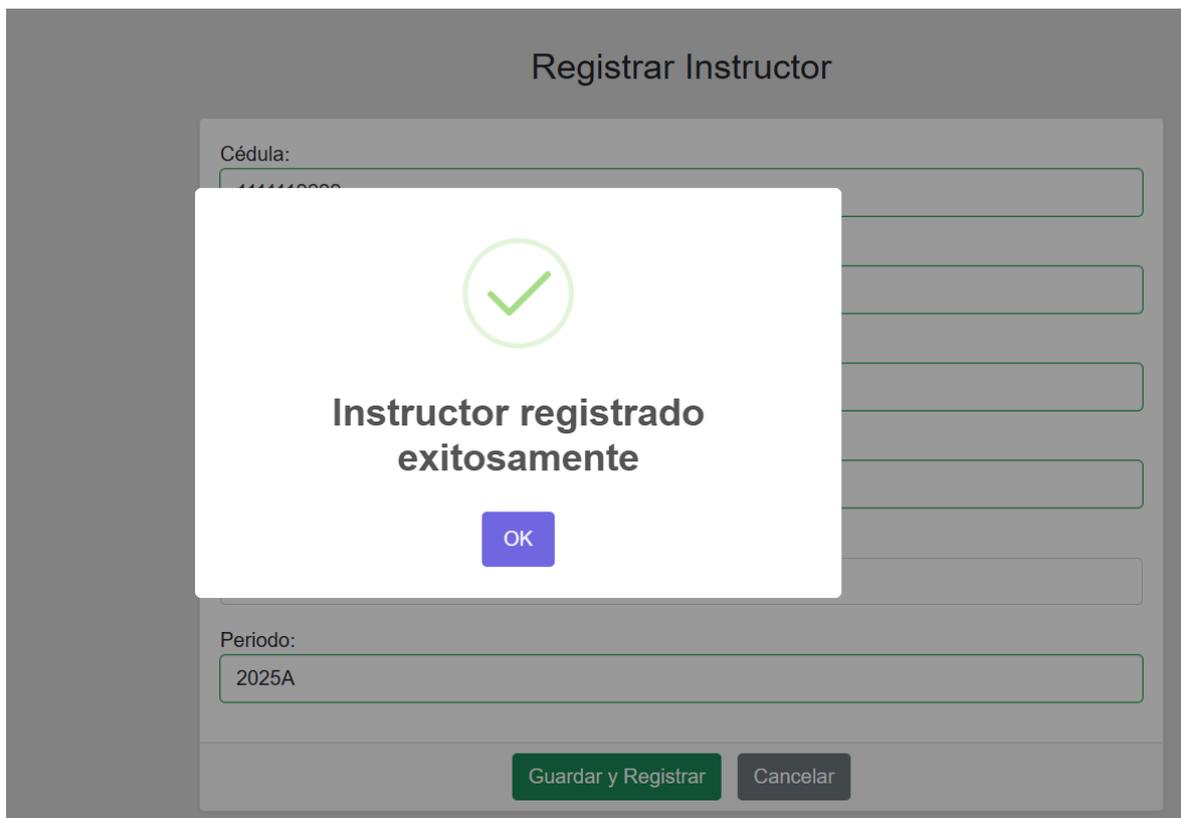


Figura 3.4: Registro exitoso de un instructor

La Figura 3.5 muestra el caso de prueba utilizado para comprobar la funcionalidad del requerimiento *HU07-Crear cursos*.

Caso de prueba: Creación exitoso de curso	
Numero: 11	Usuario: Administrador
Historia de usuario relacionada: HU07-Crear cursos	
Condiciones iniciales:	Acceso al sistema como administrador
Procedimiento:	<ol style="list-style-type: none"> 1. Navegar a la interfaz de cursos 2. Seleccionar la opción "crear curso" 3. Ingresar la información relacionada al nuevo curso: nombre, tipo de curso, horas del curso 4. Seleccionar la modalidad del curso 5. Hacer clic en "crear curso"
Resultado esperado:	El sistema muestra un mensaje de confirmación y el curso aparece en el listado de cursos El curso nuevo es creado en el sistema
Resultado obtenido:	Se obtuvo el resultado es el esperado
Cumple:	Si

Figura 3.5: Caso de prueba para la creación exitosa de un curso

Después de completar correctamente la información solicitada en el formulario de creación, se evidencia el éxito del caso de prueba *Creación exitosa de curso* en la Figura 3.6 y en la Figura 3.7

The image shows a web interface for creating a new course. The form fields are filled with the following data:

- Nombre del curso: Blazor Avanzado
- Tipo del curso: Educativo
- Horas del curso: 150
- Modalidad: VIRTUAL

 A green button labeled "Crear Curso" is visible at the bottom left. Overlaid on the form is a white dialog box with a green checkmark icon, the text "Curso creado con éxito", and a blue "OK" button.

Figura 3.6: Creación exitosa de curso

Cursos Disponibles

[Crear Nuevo Curso](#)

ID	Nombre	Tipo	Horas	Modalidad	Acciones
1	INCLUSION DIGITAL	Educativo	55	VIRTUAL	Ver Periodos Editar Eliminar
2	Blender	Software 3D	40	PRESENCIAL	Ver Periodos Editar Eliminar
3	JavaScript	Programacion	40	PRESENCIAL	Ver Periodos Editar Eliminar
4	Blazor	Programacion	40	PRESENCIAL	Ver Periodos Editar Eliminar
5	Curso Nuevisimo	Educativo	55	VIRTUAL	Ver Periodos Editar Eliminar
6	Nuevo	Prueba	100	VIRTUAL	Ver Periodos Editar Eliminar
7	Blazor Avanzado	Educativo	150	VIRTUAL	Ver Periodos Editar Eliminar

Figura 3.7: Curso creado en listado de cursos

La Figura 3.8 muestra el caso de prueba utilizado para comprobar la funcionalidad del requerimiento *HU13-Repositorio de Documentos*.

Caso de prueba: Carga de documentos al repositorio	
Numero: 23	Usuario: Instructor
Historia de usuario relacionada: HU13-Repositorio de Documentos	
Condiciones iniciales:	Acceso al sistema como instructor
Procedimiento:	<ol style="list-style-type: none"> 1. Navegar a la interfaz de repositorio 2. Seleccionar la opción "subir archivos" 3. Hacer clic en el enlace para el inicio de sesión en el dominio "iniciar sesión" 4. Iniciar sesión en el dominio 5. Hacer clic en el botón "elegir archivo" 6. Elegir el archivo para subir 7. Hacer clic en el botón "subir archivo"
Resultado esperado:	Se sube el archivo en el repositorio de SharePoint configurado Se muestra un mensaje de carga exitosa El sistema redirige a la página carga de archivos
Resultado obtenido:	Se obtuvo el resultado es el esperado
Cumple:	Si

Figura 3.8: Caso de prueba para la carga exitosa de documentos al repositorio

La comprobación del éxito del caso de prueba *Carga de documentos al repositorio* se puede visualizar en la Figura 2.45 que muestra la carga exitosa de un archivo al repositorio de SharePoint.

En definitiva, los casos de prueba reflejan que el sistema está haciendo lo que se espera que haga, pues cada uno de ellos cumple con los resultados esperados.

3.2. Pruebas de usabilidad

Dado que la usabilidad es un atributo de calidad importante, se decidió medirla en el componente desarrollado. Esto con el objetivo de asegurar que una persona con experiencia promedio en el uso de aplicaciones web pueda utilizarlo sin mayores complicaciones [31] y, sobre todo, cumplir con el propósito para el que fue diseñado.

Para este fin, se utilizó un cuestionario del tipo SUS (Escala de Usabilidad del Sistema), el cual ha demostrado ser un método de evaluación confiable y válido para medir la facilidad de uso percibida de una aplicación de software, con una alta validez psicométrica [32]. El cuestionario consta de 10 preguntas sobre la facilidad de uso del sistema, que los usuarios participantes deben responder utilizando una escala Likert de 5 puntos (donde 1 indica total desacuerdo y 5 total acuerdo) para proporcionar retroalimentación sobre la simplicidad de uso de la aplicación según su percepción. Además, se utiliza una técnica para convertir las puntuaciones en una escala general de 0 a 100. Los resultados se interpretan como una combinación global de todas las preguntas, por lo que se siguió la recomendación de usar porcentajes para evaluarlos [33].

En cuanto a los participantes, se empleó un grupo de 15 personas divididas en tres grupos de 5: uno para las funcionalidades de administrador, otro para las funcionalidades de estudiante, y el tercero para las funcionalidades de instructor. De esta forma, se aseguró contar con la cantidad óptima de participantes recomendada por Nielsen [34] para cada rol. Además, se consideraron las edades típicas de los instructores (estudiantes universitarios), los estudiantes beneficiarios (adultos, adultos mayores y docentes de educación básica), y los administradores potenciales del sistema (estudiantes y docentes universitarios) para seleccionar a los participantes de las pruebas de usabilidad, abarcando edades desde los 20 hasta los 50 años.

A continuación, se presentan las tareas a realizar para cada grupo de participantes.

Tabla 3.1: Detalle de tareas y subtareas para el rol de administrador

Tarea	Subtarea
T1. Registrar un Nuevo Estudiante	<ol style="list-style-type: none"> 1. Navegue hacia la sección de estudiantes 2. Haga clic en el botón "Agregar Nuevo Estudiante" 3. Ingrese la información del estudiante, puede utilizar su propia información 4. Seleccione el curso "Inclusión digital" y el período "2025A" de las opciones disponibles. 5. Haga clic en "Guardar y Matricular" para registrar el estudiante.
T2. Crear un nuevo curso	<ol style="list-style-type: none"> 1. Navegue a la sección de creación de cursos en la interfaz de administración 2. Ingrese un nombre para el nuevo curso 3. Ingrese la palabra "Prueba" en el tipo de curso 4. Coloque la cantidad de 10 en horas de curso 5. Seleccione la modalidad "virtual" para el curso 6. Presione el botón "Crear Curso"
T3. Crear un nuevo periodo	<ol style="list-style-type: none"> 1. Navegue a la sección de lista de cursos 2. Ubique el curso que creó en el paso anterior y seleccione la opción "Ver periodos" 3. Seleccione la opción "Crear nuevo periodo" 4. Ingrese la información para la creación del período 5. Haga clic en el botón "Crear periodo"

<p>T4. Cree nuevos grupos con asignaciones automáticas</p>	<ol style="list-style-type: none"> 1. Navegue a la sección de grupos 2. Seleccione la opción "Crear grupos" 3. Seleccione el curso "Inclusión digital" 4. Seleccione el periodo "2025A" 5. Ingrese 1 como número de estudiantes por grupo 6. Ingrese 3 como número máximo de grupos por instructor 7. Ingrese 2 como horas por clase 8 Haga clic en el botón "Crear grupos" 9. Visualice los grupos creados, el instructor asignado al grupo y los estudiantes en la sección de pre visualización 10. Haga clic en el botón "Guardar grupos"
<p>T5. Cree nuevos grupos de manera manual</p>	<ol style="list-style-type: none"> 1. Navegue a la sección de grupos 2. Seleccione la opción "Crear grupos" 3. Encuentre y seleccione el botón "Crear grupos manualmente" 4. Seleccione el curso "Inclusión digital" 5. Seleccione el periodo "2025A" 6. Seleccione como instructor a "Luis Fernandez" 7. Seleccione como estudiantes a su persona, creada en la tarea 1; y a "Roberto Paredes" 8. Agregue los horarios: lunes, de 8:00 a 10:00; y miércoles de 14:00 a 16:00

Tabla 3.2: Detalle de tareas y subtareas para el rol de estudiante

Tarea	Subtarea
T1. Ingresar disponibilidad horaria	<ol style="list-style-type: none"> 1. Navegue hacia la sección de cursos 2. Elija la opción "Cursos disponibles" 3. Seleccione el curso "Inclusión digital" 4. Ingrese dos horarios en días diferentes 5. Haga clic en "Guardar horarios"

Tabla 3.3: Detalle de tareas y subtareas para el rol de instructor

Tarea	Subtarea
T1. Registrar en un curso	<ol style="list-style-type: none"> 1. Navegue hacia la sección de "Cursos" 2. Elija la opción "Registrarse en curso" 3. Escoja el curso "Inclusión digital" 4. Ingrese dos horarios en días diferentes 5. Haga clic en "Guardar horarios"
T2. Ver la información de grupo asignado	<ol style="list-style-type: none"> 1. Navegue a la sección de cursos 2. Elija la opción "Mis cursos" 3. Escoja el curso "Inclusión digital" 4. Visualice los detalles del curso

Después de completar las tareas, se proporcionó a los participantes la encuesta SUS, descrita anteriormente. Esta encuesta consta de las preguntas que se muestran en la Figura 3.9.

Num	Pregunta
1	Creo que me gustaría usar este sistema con frecuencia.
2	Encontré el sistema innecesariamente complejo.
3	Pensé que el sistema era fácil de usar.
4	Creo que necesitaría la ayuda de una persona técnica para poder usar este sistema.
5	Encontré que las diversas funciones en este sistema estaban bien integradas.
6	Pensé que había demasiada inconsistencia en este sistema.
7	Imagino que la mayoría de las personas aprenderían a usar este sistema muy rápidamente.
8	Encontré el sistema muy engorroso de usar.
9	Me sentí muy confiado usando el sistema.
10	Necesité aprender muchas cosas antes de poder manejarlo con este sistema.

Figura 3.9: Preguntas del cuestionario de usabilidad del sistema SUS

Puntuación de usabilidad

Para obtener el puntaje de usabilidad del sistema a partir de las respuestas del cuestionario SUS se sigue un proceso de algunos pasos. Primero, se recopilan las respuestas de cada participante para las 10 preguntas de la encuesta, valoradas en una escala de 1 a 5. Luego, se recodifican las respuestas de la siguiente forma:

- Para las preguntas impares (1, 3, 5, 7, 9), se resta 1 del valor de la respuesta
- Para las preguntas pares (2, 4, 6, 8, 10), se resta el valor de la respuesta de 5.

Posteriormente, se suman todos los valores recodificados para cada participante. Para finalizar, el total obtenido se multiplica por 2.5, lo que convierte la puntuación en una escala de 0 a 100. Este resultado representa el porcentaje de usabilidad del sistema, proporcionando

una medida cuantitativa de la facilidad de uso percibida por los usuarios [32].

En el caso de la encuesta aplicada, se calculó el puntaje SUS individual para las respuestas de cada participante, para posteriormente sacar el puntaje promedio de usabilidad tras sumar los puntajes individuales y dividirlos para el número de participantes, como se muestra en la fórmula 3.1.

$$\text{Promedio SUS} = \frac{77,5 + 75,0 + 75,0 + 62,5 + 80,0 + 72,5 + 70,0 + 80,0 + 67,5}{15} + \frac{82,5 + 72,5 + 82,5 + 77,5 + 47,5 + 82,5}{15} \quad (3.1)$$

$$\text{Promedio SUS} = \frac{1105,0}{15} \approx 73,67 \quad (3.2)$$

Luego, el puntaje de usabilidad promedio para el componente administrativo es **73.67**, lo que indica que los usuarios perciben la aplicación con una usabilidad “buena” y aceptable según la clasificación por rangos de la puntuación SUS [35] que se visualiza en la Figura 3.10. A pesar de ser un valor aceptable, la retroalimentación proporcionada da a entender que se debe mejorar en cuanto a la usabilidad del componente.

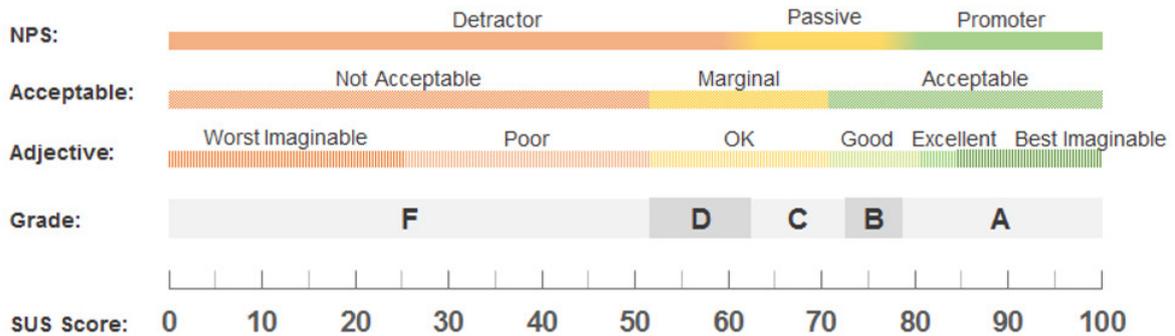


Figura 3.10: Rangos de puntuación SUS

[36]

Capítulo 4

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

El componente de administración de LudoLab ConnectSys desarrollado ofrece un control eficiente y centralizado de los cursos y participantes de los programas de capacitación de LudoLab. Cada una de sus funcionalidades fueron concebidas con la finalidad de agilizar significativamente los procesos académicos, mejorando tanto la administración de la información de los estudiantes como la supervisión del trabajo de los instructores. La implementación del módulo ha cumplido con el objetivo general de proporcionar una herramienta con la cual se facilite la gestión académica y administrativa de los programas educativos del laboratorio.

La aplicación de Kanban como metodología de desarrollo del software permitió asegurar una gestión eficiente y adaptativa del proyecto. Dado a su característica principal, que es la visualización clara del flujo de trabajo, fue posible obtener una mejor gestión de las tareas y una rápida adaptación a los cambios, además de mejorar la comunicación entre los miembros del equipo; dando como resultado un desarrollo más coherente, ágil y alineado con los objetivos del proyecto.

El submódulo desarrollado para la creación y configuración de cursos proporciona a los administradores interfaces de fácil uso, las cuales permiten realizar estos procesos de manera rápida, asequible y eficiente. De esta forma, se simplifica la gestión de los cursos, y se facilita la planificación y organización académica de los programas de capacitación de LudoLab.

La implementación del submódulo para la inscripción y matriculación de estudiantes permite optimizar la administración de los datos estudiantiles y mejorar la eficiencia en la gestión académica. Además, los formularios diseñados e implementados representan una herramienta para agilizar el proceso de inscripción, lo cual reduce el tiempo y esfuerzo requerido para registrar a los estudiantes en los cursos de capacitación en comparación a la forma que tenían de hacerlo.

La implementación de un submódulo para almacenar la información que respalde el trabajo realizado durante los cursos permite mantener un registro documental organizado y accesible. Este repositorio facilita el seguimiento y evaluación de los avances y logros de los cursos, además de la centralización de archivos derivados del proceso educativo, proporcionando una base de datos para los mismos.

Las pruebas de funcionalidad realizadas demostraron que el componente cumple con todos los requerimientos funcionales establecidos. Además, la evaluación de usabilidad mediante la encuesta SUS arrojó un puntaje de 73,67; lo cual indican que el sistema tiene un nivel de usabilidad aceptable según la percepción de los usuarios. Sin embargo, cabe recalcar que existen espacios para implementar mejoras en cuanto a la experiencia de uso y más aún si se tienen vistas a un uso masivo de la herramienta. La retroalimentación detallada recogida ha sido muy valiosa para identificar campos de mejora y garantizar que el sistema sea intuitivo y eficiente para todos los usuarios.

4.2. Recomendaciones

En cuanto al uso y experimentación para su desarrollo, se recomienda continuar utilizando la metodología Kanban para futuras mejoras en el sistema. Gracias a su enfoque ágil, Kanban ha demostrado ser un método eficiente y adaptable a las necesidades cambiantes de los usuarios. Aunque su aplicación es menos común en comparación con otras metodologías y marcos de trabajo, el uso de Kanban puede ser particularmente beneficioso para aquellos que son novatos en este campo, ya que proporciona una guía clara sobre el funcionamiento y los principios del agilismo. Además, debido a su naturaleza poco prescriptiva, facilita su adaptación a equipos con diversas características y necesidades.

Se recomienda considerar la expansión de las funcionalidades del sistema para cubrir otras necesidades académicas y administrativas de LudoLab. Como se mencionó en el alcance del trabajo, este sistema puede servir como una base sólida para futuras implementaciones. Entre las posibles mejoras se incluye la optimización del submódulo de repositorio,

que debido a dependencias de terceros, no pudo ser expandido más allá de lo implementado inicialmente. Además, se sugiere la integración con plataformas de aprendizaje en línea, la gestión de eventos académicos y la incorporación de herramientas avanzadas de análisis y reporte. Ampliar las capacidades del sistema no solo aumentará su valor y utilidad para el laboratorio, sino también para otras instituciones que puedan beneficiarse de estas mejoras.

En cuanto a la usabilidad del sistema, se recomienda realizar mejoras en las interfaces de usuario para futuras adiciones funcionales. Asimismo, es aconsejable continuar realizando pruebas de usabilidad y funcionalidad de manera regular para asegurar que el sistema cumpla con los futuros requisitos y sea fácil de usar. La evaluación continua de la usabilidad mediante herramientas como la encuesta SUS, junto con la implementación de casos de prueba, ayudará a identificar áreas de mejora y garantizará una experiencia de usuario satisfactoria.

Capítulo 5

REFERENCIAS BIBLIOGRÁFICAS

- [1] C. Systems, «La transformación digital de la educación, un puente al futuro de Colombia,» *Cisco Systems*, 2022. dirección: https://www.cisco.com/c/dam/global/es_co/products/pdfs/whitepaper-educacion.pdf.
- [2] S. Miller, «What is a Student Information System?Modernize Higher Education with Student Information Systems,» *Software Advice Resources*, 2023. dirección: <https://www.softwareadvice.com/resources/what-is-a-student-information-system/>.
- [3] K. G. C. Chávez, «Implementación de un Sistema Web para la Gestión del Proceso Académico en la Institución Educativa Ricardo Palma,» Tesis de grado, Universidad César Vallejo, 2018. dirección: https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/28343/Chilingano_CHKG..pdf?sequence=1&isAllowed=y.
- [4] R. Molano y A. Dalila, «La gestión educativa: Hacia la optimización de la formación docente en la educación superior en Colombia,» *Sophia*, vol. 12, n.º 1, págs. 55-70, 2016, ISSN: 1794-8932.
- [5] M. A. G. Maguiño, S. L. R. Vela, R. A. R. Lozano y G. F. G. Mendocilla, «Tecnología en el proceso educativo: nuevos escenarios,» *Revista Venezolana de Gerencia*, vol. 25, n.º 92, 2020, ISSN: 1315-9984. dirección: <https://www.redalyc.org/articulo.oa?id=29065286032>.
- [6] Ellucian, «¿Qué es un Sistema de Información Estudiantil para Educación Superior?» *Ellucian Ideas*, 2024. dirección: <https://www.ellucian.com/es/ideas/que-es-un-sistema-de-informacion-estudiantil-para-educacion-superior>.

- [7] P. Govender, «Understanding the Importance of a Student Information System (SIS) in Education,» *Adapt IT Education*, 2023. dirección: <https://education.adaptit.tech/blog/understanding-the-importance-of-a-student-information-system-sis-in-education/>.
- [8] T. Creatrix, «Top 6 Advantages of Student Information Systems in Higher Ed,» *Creatrix Campus*, 2021. dirección: <https://www.creatrixcampus.com/blog/advantages-of-student-information-system>.
- [9] A. Shanganlall, «The Role of Student Information Systems in Higher Education,» *Classter*, 2024. dirección: <https://www.classter.com/blog/edtech/the-role-of-student-information-systems-in-higher-education/>.
- [10] U. A. Shaikh, «Benefits Of Student Information System: Transforming Education with QualCampus SIS,» *QualCampus*, 2023. dirección: <https://qualcampus.com/benefits-of-student-information-system-sis/>.
- [11] M. W. Docs, «SPA (Single-page application),» *MDN Web Docs Glossary*, 2024. dirección: <https://developer.mozilla.org/en-US/docs/Glossary/SPA>.
- [12] T. Ohno, *Toyota Production System: Beyond Large-Scale Production*. Portland, OR: Productivity Press, 1988, Accessed via VNBIMS DSpace on 2024-07-24. dirección: <http://dspace.vnbrims.org:13000/jspui/bitstream/123456789/4694/1/Toyota%20Production%20System%20Beyond%20Large-Scale%20Production.pdf>.
- [13] D. Mitra e I. Mitrani, «Analysis of a Kanban Discipline for Cell Coordination in Production Lines. I,» *Management Science*, vol. 36, n.º 12, págs. 1548-1566, 1990, ISSN: 00251909, 15265501. dirección: <http://www.jstor.org/stable/2661550>.
- [14] M. D. A. Serna, L. F. C. Zapata y J. A. Z. Cortes, «Mejoramiento de procesos de manufactura utilizando Kanban,» *Revista Ingenierías Universidad de Medellín*, vol. 14, n.º 27, págs. 221-234, 2015. dirección: <http://www.scielo.org.co/pdf/rium/v14n27/v14n27a14.pdf>.
- [15] D. J. Anderson, *Kanban: Successful Evolutionary Change for Your Technology Business*. Sequim, Washington: Blue Hole Press, 2010, ISBN: 978-0-9845214-2-9.
- [16] D. I. Guzmán, U. C. Islas, C. P. Corona y B. E. P. Méndez, «Metodología ágil Scrumban en el proceso de desarrollo y mantenimiento de software de la norma MoProSoft,» *Research in Computing Science*, vol. 79, págs. 97-107, 2014. dirección: https://rcs.cic.ipn.mx/2014_79/Metodologia%20agil%20Scrumban%20en%20el%20proceso%20de%20desarrollo%20y%20mantenimiento%20de%20software%20de%20la%20norma%20MoProSoft.pdf.

20de%20desarrollo%20y%20mantenimiento%20de%20software%20de%20la%20norma.pdf.

- [17] D. J. Anderson y A. Carmichael, *Kanban Esencial Condensado*. Seattle, Washington: Lean Kanban University Press, 2016, ISBN: 978-0-9845214-2-5.
- [18] Microsoft, *Introducción a .NET*, 2024. dirección: https://learn.microsoft.com/es-es/dotnet/core/introduction?WT.mc_id=dotnet-35129-website.
- [19] S. L. Daniel Roth Rick Anderson, *Introduction to ASP.NET Core*, 2024. dirección: <https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-8.0&viewFallbackFrom=aspnetcore-8.0>.
- [20] Microsoft, *Blazor de ASP.NET Core*, 2024. dirección: <https://learn.microsoft.com/es-es/aspnet/core/blazor/?view=aspnetcore-8.0>.
- [21] Microsoft, *¿Qué es SQL Server?* 2024. dirección: <https://learn.microsoft.com/es-es/sql/sql-server/what-is-sql-server?view=sql-server-ver16>.
- [22] Microsoft, *¿Qué es el IDE de Visual Studio?* 2024. dirección: <https://learn.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2022>.
- [23] Microsoft, *¿Qué es Git?* Accedido: 26-julio-2024, oct. de 2023. dirección: <https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>.
- [24] GitHub, *Acerca de GitHub y Git*, Accedido: 26-julio-2024, jul. de 2024. dirección: <https://docs.github.com/es/get-started/start-your-journey/about-github-and-git>.
- [25] Figma, *What is Figma?* Accedido: 26-julio-2024, jul. de 2024. dirección: <https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma>.
- [26] Microsoft, *¿Qué es SharePoint?* Accedido: 26-julio-2024, jul. de 2023. dirección: <https://support.microsoft.com/es-es/office/-qu%C3%A9-es-sharepoint-97b915e6-651b-43b2-827d-fb25777f446f>.
- [27] H. Parker, *What is ClickUp Used for and How Does It Work?* Accedido: 26-julio-2024, oct. de 2023. dirección: <https://clickup.com/blog/what-is-clickup-used-for/>.
- [28] Escuela Politécnica Nacional, «Proyecto de Inclusión Digital graduó a 62 beneficiarios,» *Escuela Politécnica Nacional*, 2024. dirección: <https://www.epn.edu.ec/proyecto-de-inclusion-digital-graduado-a-62-beneficiarios/>.

- [29] Microsoft, *Microsoft Graph Overview*, Accessed: 2024-07-19, 2024. dirección: <https://learn.microsoft.com/es-es/graph/overview>.
- [30] IBM, *Software Testing*, Accessed: 2024-07-20, 2024. dirección: <https://www.ibm.com/mx-es/topics/software-testing>.
- [31] S. Krug, *No me hagas pensar: una aproximación a la usabilidad en la web*. Pearson educacion, 2006.
- [32] Y. K. Macías Guamangate, J. L. Miranda Caisaluisa y V. d. C. Tapia Cerda, «Medición de usabilidad y portabilidad de una Aplicación Web desarrollada con tecnología PWA,» *Conciencia Digital*, vol. 4, n.º 4, págs. 6-27, 2021. DOI: 10.33262/concienciadigital.v4i4.1882.
- [33] A. de Castro Martín, *Herramienta de Soporte para la Evaluación Subjetiva de la Usabilidad Mediante SUS – System Usability Scale*, Bachelor Thesis, Tutor: José Antonio Macías Iglesias, Madrid, Spain, mayo de 2016.
- [34] J. Nielsen, *Why You Only Need to Test with 5 Users*, Accessed: 2024-07-21, 2000. dirección: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.
- [35] A. Bangor, P. Kortum y J. Miller, «Determining what individual SUS scores mean: Adding an adjective rating scale,» *Journal of Usability Studies*, vol. 4, n.º 3, págs. 114-123, 2009. dirección: <https://uxpajournal.org/determining-what-individual-sus-scores-mean-adding-an-adjective-rating-scale/>.
- [36] S. G. Artigas, *Cinco formas de interpretar un SUS*, Accessed: 2024-07-24, 2018. dirección: <https://torresburriel.com/weblog/cinco-formas-de-interpretar-un-sus/>.

ANEXOS

ANEXO A: [DOCUMENTO DETALLADO DE HISTORIAS DE USUARIO](#)

ANEXO B: [MOCKUPS DEL SISTEMA EN FIGMA](#)

ANEXO C: [CASOS DE PRUEBA PARA EL COMPONENTE](#)

ANEXO D: [TABLERO KANBAN UTILIZADO PARA EL DESARROLLO EN CLICKUP](#)

ANEXO E: [REPOSITORIO EN GITHUB DEL PROYECTO FRONTEND](#)

ANEXO F: [REPOSITORIO EN GITHUB DEL PROYECTO BACKEND](#)

ANEXO G: [REPOSITORIO EN GITHUB DEL DIRECTORIO DE ARCHIVOS](#)

ANEXO H: [REPOSITORIO EN GITHUB DEL SERVICIO DE AUTENTICACIÓN](#)