

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Software

Trabajo de grado previo a la obtención del título de Ingeniero de Software presentado ante la ilustre Universidad Técnica del Norte.

Tema:

“Implementación de la Fase 2 del servicio ‘Portafolio Docente’ en la plataforma móvil de la Universidad Técnica del Norte, empleando el marco de trabajo SCRUM y validando su calidad en uso conforme a la norma ISO/IEC 25022”



Autor: Diego David Recalde Varela

Director: José Antonio Quiña Mera

Ibarra – Ecuador

2026

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

Por medio del presente y en cumplimiento con el Art. 144 de la Ley de Educación Superior, dejo constancia de mi voluntad de que el presente trabajo de titulación sea entregado a la Universidad Técnica del Norte y publicado en el Repositorio Digital Institucional. Para tal fin, dispongo la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1004548960		
APELLIDOS Y NOMBRES:	Recalde Varela Diego David		
DIRECCIÓN:	Ibarra		
EMAIL:	ddrecaldev@utn.edu.ec / diegodavidrecalde@gmail.com		
TELÉFONO FIJO:	062611294	TELÉFONO MÓVIL:	0967451554

DATOS DE LA OBRA	
TÍTULO:	Implementación de la Fase 2 del servicio 'Portafolio Docente' en la plataforma móvil de la Universidad Técnica del Norte, empleando el marco de trabajo SCRUM y validando su calidad en uso conforme a la norma ISO/IEC 25022
AUTOR:	Recalde Varela Diego David
FECHA:	3/03/2026
PROGRAMA:	☒ PREGRADO ☐ POSGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO EN SOFTWARE
DIRECTOR:	Ing. Quiña Mera José Antonio, Ph.D.

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 3 días del mes de marzo de 2026

EL AUTOR:

Diego David Recalde Varela
C.C.: 1004548960

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 3 de marzo de 2026

Ing. José Antonio Quiña Mera, Ph.D.
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

Ing. José Antonio Quiña Mera, Ph.D.
Director de Trabajo de Grado
C.C.: 1002322384

DEDICATORIA

A mi mamá Isabel, por ser el pilar más grande en mi vida. Por todo el sacrificio y el inmensurable esfuerzo que realiza cada día para apoyarme y acompañarme. Por aconsejarme cuando más lo necesitaba, por cuidarme en mis momentos de debilidad y por siempre estar a mi lado, a pesar de las circunstancias adversas. Valores como su inquebrantable fortaleza, su constante responsabilidad y su amor incondicional han sido mi inspiración para llegar a ser el hombre y profesional que soy. No hay palabras suficientes para expresar el agradecimiento y amor que siento. Gracias, Mamá.

A mi papá Galo, por su guía constante y su apoyo incondicional a lo largo de mi vida. Por saber corregirme y orientarme cuando cometía errores, por animarme y levantarme cuando estaba decaído, por comprenderme y aceptarme tal como soy. Gracias a su enorme esfuerzo, sabiduría y sentido del humor he podido avanzar con firmeza en esta etapa de mi vida. Gracias, Papá.

A mi hermano Galo, quien, pese a que ya no vivimos juntos, siempre ocupa un lugar especial en mi corazón, con gran cariño y admiración. Su increíble resiliencia, su incansable perseverancia y su independencia demostradas en su camino para convertirse en un gran doctor han sido una fuente de inspiración en mi proceso a convertirme en profesional.

A mis abuelitos, quienes con su enorme cariño y sus sabias enseñanzas han sido fundamentales para aprender actuar con humildad, responsabilidad y compasión en el camino de la vida.

A mi abuelita María, que recientemente partió de este mundo, pero permanece viva en mi corazón. Ella me acompañó desde mis primeros pasos; su amor y su legado seguirán siendo mi fuerza durante toda mi vida. Dedico este trabajo con infinito cariño y profunda nostalgia.

A mi familia, por su grata compañía y amor en cada paso de este camino para convertirme en ingeniero.

A mi gatito Coby, por su amor sincero y por ser mi compañero de vida y estudio. Él ha sido mi soporte emocional y estuvo conmigo desde el primer día en que empecé esta desafiante etapa.

AGRADECIMIENTO

Agradezco a la Universidad Técnica del Norte, por brindarme el espacio y acceso a educación de calidad, por darme la oportunidad de aprender de la mano de docentes rigurosos que han sabido impartir su gran dominio sobre la Ingeniería de Software, así como compartir valiosas enseñanzas y experiencias que me han servido profundamente para llegar a ser un buen profesional.

A las amistades que he formado en estos años, gracias por su acompañamiento, ayuda y vivencias compartidas en esta tan memorable etapa de mi vida.

A mi tutor de tesis, el Ing. Antonio Quiña, por haber confiado en mí y haberme dado la gran oportunidad de ser parte del proyecto institucional UTN-Móvil. Su mentoría, consejos y acompañamiento han sido cruciales para que el desarrollo e implementación de este trabajo de titulación pueda ser culminado con éxito. Estoy profundamente agradecido de ser parte de este equipo y haber hecho mi aporte.

Al personal de la Dirección de Desarrollo Tecnológico e Informático, en especial los ingenieros Franklin Enriquez y Jorge Caraguay, quienes gestionaron y me brindaron todas las facilidades para el desarrollo de este proyecto, así como su acompañamiento y ayuda ante cualquier duda que surgió a lo largo de este retador proceso.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
Planteamiento del problema	1
Objetivos.....	2
Objetivo General.....	2
Objetivos Específicos	2
Alcance	2
Metodología.....	4
Justificación	5
Justificación Tecnológica	5
Justificación Social	5
Contexto de la Investigación.....	5
I MARCO TEÓRICO	7
1.1 Sistema informático integrado universitario (SIIU)	7
1.1.1 Portafolio Docente UTN	7
1.1.2 Implementación previa: funcionalidades desarrolladas en la Fase 1	7
1.1.3 Infraestructura tecnológica del SIIU.....	8
1.1.4 Oracle APEX	8
1.1.5 Base de datos Oracle.....	8
1.2 Arquitectura de microservicios en el backend	9
1.2.1 Principios de la arquitectura de microservicios	9
1.2.2 Comunicación entre microservicios y APIs RESTful.....	10
1.2.3 Lenguaje de programación TypeScript.....	10
1.2.4 Framework NestJS para desarrollo de microservicios	11
1.3 Aplicaciones móviles	13
1.3.1 Aplicaciones móviles en entornos universitarios	13

1.3.2	Tendencias del uso de plataformas móviles	14
1.3.3	Tipos de aplicaciones móviles	14
1.3.4	Aplicaciones móviles multiplataforma.....	15
1.3.5	Flutter como framework multiplataforma	15
1.3.6	Lenguaje de programación Dart	17
1.4	Metodología ágil de desarrollo	18
1.4.1	Fundamentos de metodologías ágiles y marco Scrum	18
1.4.2	Roles	18
1.4.3	Artefactos	19
1.4.4	Fases de desarrollo: Pre-Juego, Juego y Post-Juego	20
1.5	Evaluación de calidad en uso del software.....	21
1.5.1	Norma ISO/IEC 25010: Modelo de calidad del producto de software	21
1.5.2	Calidad en uso según la norma ISO/IEC 25010.	22
1.5.3	Norma ISO/IEC 25022: Medición de la calidad en uso	22
1.5.4	Escalas e instrumentos de medición de calidad en uso	23
1.6	Trabajos relacionados	23
II	DESARROLLO	27
2.1	Etapas de pre-juego.....	28
2.1.1	Definición del equipo Scrum	28
2.1.2	Levantamiento de requisitos	28
2.1.2.1	Historias de Usuario	29
2.1.3	Product backlog.....	44
2.2	Etapas de juego.....	45
2.2.1	Sprint 0	45
2.2.1.1	Definición de arquitectura de microservicios	45
2.2.1.2	Entorno de desarrollo	46
2.2.1.3	Estructura del proyecto backend.....	47

2.2.1.4	Estructura del proyecto frontend	49
2.2.2	Sprint 1	52
2.2.2.1	Planificación del sprint 1.....	52
2.2.2.2	Incremento del sprint 1	53
2.2.2.3	Revisión del sprint 1	66
2.2.2.4	Retrospectiva del sprint 1	67
2.2.3	Sprint 2	68
2.2.3.1	Planificación del sprint 2.....	68
2.2.3.2	Incremento del sprint 2.....	69
2.2.3.3	Revisión del sprint 2	81
2.2.3.4	Retrospectiva del sprint 2.....	84
2.2.4	Sprint 3	84
2.2.4.1	Planificación del sprint 3.....	84
2.2.4.2	Incremento del sprint 3.....	85
2.2.4.3	Revisión del sprint 3	91
2.2.4.4	Retrospectiva del sprint 3.....	93
2.2.5	Sprint 4	94
2.2.5.1	Planificación del sprint 4.....	94
2.2.5.2	Incremento del sprint 4.....	95
2.2.5.3	Revisión del sprint 4	103
2.2.5.4	Retrospectiva del sprint 4.....	105
2.3	Etapas de post-juego	105
III	VALIDACIÓN DE RESULTADOS.....	107
3.1	Definición del modelo de calidad en uso	107
3.2	Medición de la calidad en uso.....	108
3.2.1	Muestra poblacional	108
3.2.2	Taller práctico.....	109
3.2.3	Encuesta SUS.....	111
3.2.4	Fiabilidad del taller práctico	112
3.2.5	Fiabilidad de la encuesta SUS.....	112

3.3	Evaluación del modelo de calidad en uso.....	113
3.3.1	Eficacia	113
3.3.2	Eficiencia	114
3.3.3	Satisfacción	115
3.4	Resultados del modelo de calidad en uso	119
CONCLUSIONES		121
RECOMENDACIONES		122
REFERENCIAS		123
ANEXOS		128
	Anexo A. Fotografías de la evaluación a docentes	128
	Anexo B. Taller práctico para evaluar la Fase 2	129
	Anexo C. Resultados del Taller Práctico	130
	Anexo D. Resultados de la encuesta SUS	131

ÍNDICE DE FIGURAS

Figura 1 Árbol de Problemas	2
Figura 2 Arquitectura de la aplicación móvil UTN-Móvil	3
Figura 3 Arquitectura de una aplicación Nest	12
Figura 4 Arquitectura de una aplicación Flutter	16
Figura 5 Diagrama del proceso Scrum.....	21
Figura 6 Divisiones de la familia de normas ISO/IEC 25000	22
Figura 7 Diagrama de fases de Scrum.....	27
Figura 8 Arquitectura de microservicios UTN-Móvil.....	46
Figura 9 Estructura del microservicio docente	47
Figura 10 Estructura de un módulo de NestJS	48
Figura 11 Estructura del cliente Flutter	50
Figura 12 Opciones de proyectos de investigación	53
Figura 13 Mis proyectos de investigación	54
Figura 14 Listado de convocatorias	54
Figura 15 Listado de proyectos por convocatoria	55
Figura 16 Filtrado de proyectos	55
Figura 17 Filtrado de convocatorias	56
Figura 18 Datos generales de proyecto	56
Figura 19 Objetivos específicos del proyecto	57
Figura 20 Áreas de investigación y ejes integrados del proyecto	57

Figura 21 Sectores de impacto del proyecto.....	58
Figura 22 Instituciones asociadas al proyecto	58
Figura 23 Beneficiarios del proyecto	59
Figura 24 Equipo de trabajo del proyecto	59
Figura 25 Estrategias de socialización del proyecto	60
Figura 26 Acciones de divulgación del proyecto.....	60
Figura 27 Estrategias de producción científica del proyecto	61
Figura 28 Infraestructura del proyecto	61
Figura 29 Documentos del proyecto	62
Figura 30 Listado de pagos	63
Figura 31 Filtrado de pagos.....	63
Figura 32 Reporte de pago generado	64
Figura 33 Nuevo ícono de crear aula virtual	65
Figura 34 Confirmación de creación de aula virtual.....	65
Figura 35 Tipos de obra de producción científica	70
Figura 36 Lista de artículos	70
Figura 37 Lista de libros	71
Figura 38 Lista de capítulos de libros.....	71
Figura 39 Lista de ponencias	72
Figura 40 Lista de obras artísticas	72
Figura 41 Filtrado de artículos.....	73

Figura 42 Datos generales de obra	74
Figura 43 Autores de obra	74
Figura 44 Documentos de obra	75
Figura 45 Acceso a foro general	76
Figura 46 Acceso a foro de actividad	76
Figura 47 Comentarios de foro	77
Figura 48 Participación en foro	78
Figura 49 Lista de control de bienes	79
Figura 50 Filtrado de control de bienes	79
Figura 51 Visualización de fotografía de activo	80
Figura 52 Tipos de reporte de control de bienes	80
Figura 53 Reporte de control de bienes generado en PDF	81
Figura 54 Lista de preguntas	86
Figura 55 Filtrado de preguntas	86
Figura 56 Opciones de pregunta	87
Figura 57 Detalle de pregunta	87
Figura 58 Formulario de pregunta sección 1	88
Figura 59 Formulario de pregunta sección 2	89
Figura 60 Formulario de importar preguntas sección 1	90
Figura 61 Formulario de importar preguntas sección 2	90
Figura 62 Opción de agregar preguntas	96

Figura 63 Listados de preguntas agregadas y disponibles de evaluación	97
Figura 64 Filtrado de preguntas disponibles	97
Figura 65 Quitar preguntas de evaluación	98
Figura 66 Agregar preguntas a evaluación	98
Figura 67 Opción de sortear preguntas.....	99
Figura 68 Configuración del sorteo	100
Figura 69 Preguntas sorteadas a cada estudiante	101
Figura 70 Acceso a revisión de evaluación.....	102
Figura 71 Revisión de evaluación online.....	103
Figura 72 Alfa de Cronbach encuesta SUS	113
Figura 73 Rango de resultados de calidad en uso	119

ÍNDICE DE TABLAS

Tabla 1 Antecedentes	5
Tabla 2 Integrantes del Equipo Scrum	28
Tabla 3 Planificación General de Sprints	29
Tabla 4 Historia de Usuario 1	29
Tabla 5 Historia de Usuario 2	30
Tabla 6 Historia de Usuario 3	30
Tabla 7 Historia de Usuario 4	31
Tabla 8 Historia de Usuario 5	31
Tabla 9 Historia de Usuario 6	32
Tabla 10 Historia de Usuario 7	33
Tabla 11 Historia de Usuario 8	34
Tabla 12 Historia de Usuario 9	35
Tabla 13 Historia de Usuario 10	36
Tabla 14 Historia de Usuario 11	36
Tabla 15 Historia de Usuario 12	37
Tabla 16 Historia de Usuario 13	38
Tabla 17 Historia de Usuario 14	39
Tabla 18 Historia de Usuario 15	39
Tabla 19 Historia de Usuario 16	40
Tabla 20 Historia de Usuario 17	41

Tabla 21 Historia de Usuario 18	42
Tabla 22 Historia de Usuario 19	43
Tabla 23 Historia de Usuario 20	43
Tabla 24 Product Backlog.....	44
Tabla 25 Sprint backlog 1	52
Tabla 26 Revisión de éxito por criterios de aceptación del sprint 1	66
Tabla 27 Sprint backlog 2.....	68
Tabla 28 Revisión de éxito por criterios de aceptación del sprint 2	81
Tabla 29 Sprint backlog 3.....	84
Tabla 30 Revisión de éxito por criterios de aceptación del sprint 3	91
Tabla 31 Sprint backlog 4	94
Tabla 32 Revisión de éxito por criterios de aceptación del sprint 4	103
Tabla 33 Modelo de Calidad en Uso	108
Tabla 34 Diseño del taller práctico	109
Tabla 35 Preguntas encuesta SUS	111
Tabla 36 Pesos de respuesta de encuesta SUS	116
Tabla 37 Resultados de la encuesta para la utilidad	116
Tabla 38 Resultados de la encuesta para la comodidad	118
Tabla 39 Resultados de calidad en uso	120

RESUMEN

La Universidad Técnica del Norte cuenta con la aplicación móvil institucional UTN-Móvil, la cual ofrece diversos servicios académicos a la comunidad universitaria; sin embargo, el servicio de "Portafolio Docente" requería la incorporación de funcionalidades complementarias para mejorar la gestión y visualización de la información académica de los docentes. El objetivo del presente trabajo fue implementar la Fase 2 del servicio "Portafolio Docente" en la plataforma móvil de la Universidad Técnica del Norte, empleando el marco de trabajo SCRUM y validando su calidad en uso conforme a la norma ISO/IEC 25022. Se realizó una investigación de tipo aplicado; para el desarrollo se empleó la metodología ágil SCRUM, estructurada en cinco sprints, utilizando NestJS como framework backend basado en microservicios y Flutter como framework frontend multiplataforma, conectados a la base de datos Oracle institucional. Se implementaron ocho funcionalidades: creación de aula virtual, banco de preguntas, evaluación en línea, foro en línea, visualización de proyectos de investigación, producción científica, control de bienes y pagos. La calidad en uso fue evaluada mediante un taller práctico y la encuesta SUS aplicados a 20 docentes, obteniendo un 88.28% de calidad en uso global, distribuido en un 42.00% de eficacia, un 24.96% de eficiencia y un 21.32% de satisfacción; resultado que se ubica en el rango satisfactorio y excede los requisitos según la norma ISO/IEC 25022. Se concluye que la implementación de la Fase 2 fue exitosa, demostrando que la aplicación móvil es efectiva, eficiente y aceptada por los docentes como una solución de calidad para la gestión académica.

Palabras clave: portafolio docente, aplicación móvil, SCRUM, calidad en uso, ISO/IEC 25022, microservicios.

ABSTRACT

The Universidad Técnica del Norte has an institutional mobile application called UTN-Móvil, which provides various academic services to the university community; however, the "Teaching Portfolio" service required the addition of complementary functionalities to improve the management and visualization of faculty academic information. The objective of this study was to implement Phase 2 of the Teaching Portfolio" service on the mobile platform of the Universidad Técnica del Norte, using the SCRUM framework and validating its quality in use according to the ISO/IEC 25022 standard. An applied research approach was adopted; for the development, the agile SCRUM methodology was employed, structured in five sprints, using NestJS as a microservices-based backend framework and Flutter as a cross-platform frontend framework, connected to the institutional Oracle database. Eight functionalities were implemented: virtual classroom creation, question bank, online assessment, online forum, visualization of research projects, scientific production, asset control, and payments. Quality in use was evaluated through a practical workshop and the SUS survey administered to 20 faculty members, yielding an overall quality in use score of 88.28%, distributed as 42.00% for effectiveness, 24.96% for efficiency, and 21.32% for satisfaction; a result that falls within the satisfactory range and exceeds the requirements established by the ISO/IEC 25022 standard. It is concluded that the implementation of Phase 2 was successful, demonstrating that the mobile application is effective, efficient, and accepted by faculty as a quality solution for academic management.

Keywords: teaching portfolio, mobile application, SCRUM, quality in use, ISO/IEC 25022, microservices.

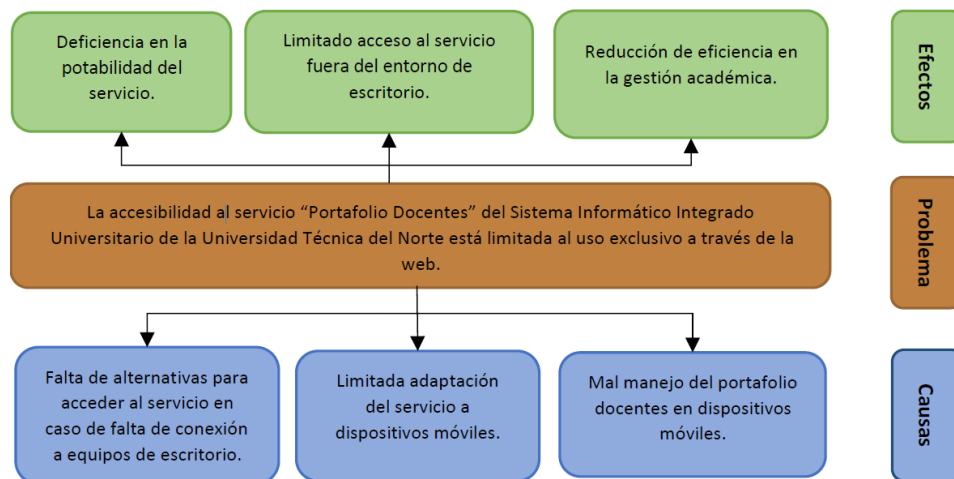
INTRODUCCIÓN

Planteamiento del problema

A pesar del uso generalizado de teléfonos inteligentes en el ámbito académico, donde en Ecuador un 92,2% de personas con educación superior poseen un teléfono inteligente [1], la Universidad Técnica del Norte (UTN) sigue dependiendo exclusivamente de su plataforma web para la gestión académica, por lo que se carece de una aplicación móvil dedicada que ofrezca una experiencia de usuario intuitiva y conveniente, tanto para docentes como para estudiantes. Según un estudio, las aplicaciones móviles universitarias ofrecen múltiples beneficios, como la optimización de funciones académicas, la reducción de la carga administrativa y la mejora en la gestión de recursos y comunicación docente [2].

La comunidad universitaria tiene a su disposición una variedad de servicios de gestión académica integrados en el Sistema Informático Integrado Universitario (SIIU), siendo uno de los más fundamentales para el desarrollo académico el servicio de “Portafolio Docente”. Sin embargo, en la actualidad, la accesibilidad a dicho servicio se ve limitada debido a su uso exclusivo a través de la web, el cual carece de un diseño responsivo por lo que la experiencia de usuario en móviles no es cómoda ni intuitiva. Ante este problema, surge la necesidad y oportunidad de desarrollar una aplicación móvil dedicada a la gestión académica de la UTN en donde se incluya el servicio de “Portafolio Docente”. La implementación del servicio móvil para docentes abordará la limitación de accesibilidad al módulo, con la finalidad de mejorar el desempeño de las actividades principales de los docentes a través de la plataforma móvil UTN.

Figura 1 Árbol de Problemas



Fuente: Propia

Objetivos

Objetivo General

Implementar la Fase 2 del servicio "Portafolio Docente" en la plataforma móvil de la Universidad Técnica del Norte, empleando el marco de trabajo SCRUM y validando su calidad en uso conforme a la norma ISO/IEC 25022.

Objetivos Específicos

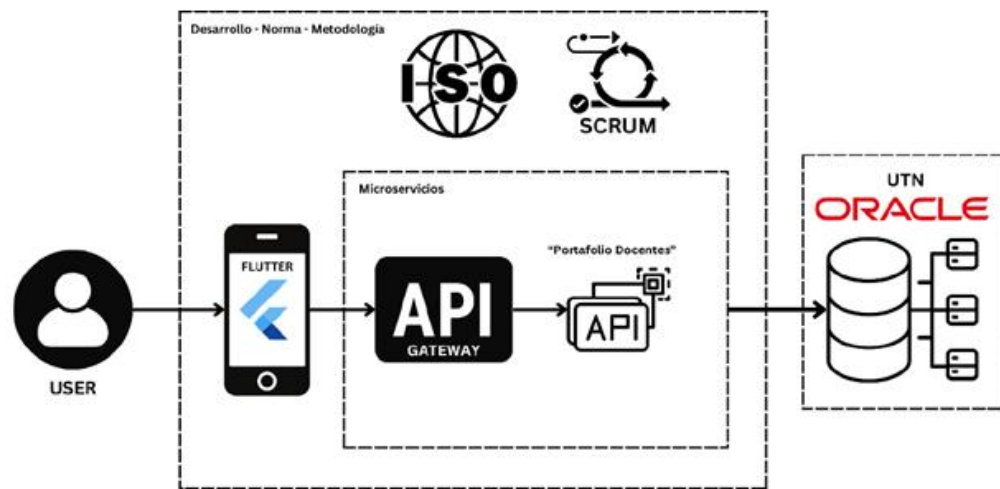
1. Establecer un marco conceptual para el estudio del desarrollo del servicio "Portafolio Docente" en la aplicación móvil.
2. Implementar las nuevas funcionalidades para el servicio de "Portafolio Docente" en la aplicación móvil a través del marco de trabajo SCRUM.
3. Evaluar la calidad de uso de la aplicación móvil para la Fase 2 del servicio "Portafolio Docente" con la norma ISO/IEC 25022.

Alcance

El presente trabajo de investigación tiene como objetivo la implementación de la Fase 2 del servicio "Portafolio Docente" para la aplicación móvil de la Universidad Técnica del Norte. En esta fase, se implementarán funcionalidades complementarias que amplíen las capacidades

actuales del servicio existente, con el objetivo de mejorar la gestión y visualización del portafolio docente. En relación con el desarrollo de la aplicación móvil, se continuará trabajando en base a la arquitectura y tecnologías definidas previamente en la Fase 1, por lo tanto, se utilizará NestJS en el backend, garantizando una arquitectura escalable y eficiente basada en microservicios. Y se utilizará Flutter en el frontend, permitiendo desarrollar una aplicación móvil compatible en múltiples plataformas (Android, iOS) con una experiencia de usuario fluida. En relación con los datos, la aplicación móvil se conectará a la base de datos Oracle ya establecida en la universidad, asegurando continuidad y compatibilidad con la información existente. Se utilizará la arquitectura de microservicios ya establecida en la Fase 1 del proyecto, como se observa en la figura 2.

Figura 2 Arquitectura de la aplicación móvil UTN-Móvil



Fuente: [3]

Las funcionalidades pendientes que se implementarán en la Fase 2 del “Portafolio Docente” son las siguientes:

- **Creación de aula virtual:** Opción para crear el aula virtual, el cual es un espacio en el que se gestionan todas las actividades de una materia dentro de un periodo específico.
- **Banco de preguntas:** Gestión de preguntas propuestas por el docente para que los estudiantes respondan en una evaluación.

- **Evaluación online:** Tipo de actividad académica integrada en la aplicación móvil en la cual el docente debe agregar o sortear preguntas a partir del banco de preguntas para así crear una evaluación online en la que participen los estudiantes.
- **Foro online:** Espacio de opinión y debate en el que el docente propone opiniones o preguntas para que los estudiantes participen.
- **Proyectos de investigación:** Visualización de trabajo estructurados científicos.
- **Producción científica:** Visualización de obras producidas por el docente, clasificadas en los siguientes tipos: artículos, libros, capítulos de libro, ponencias y obras artísticas.
- **Control de bienes:** Visualización de inventario de activos y bienes a cargo del docente.
- **Pagos:** Visualización de las transacciones hechas por el docente.

Metodología

Con el fin de implementar la Fase 2 del “Portafolio Docente” en la aplicación móvil de la UTN, se realizará un trabajo de investigación de tipo aplicado. Para ello, se realizarán las siguientes actividades de acuerdo con cada objetivo:

1. Para establecer el marco teórico se realizará una revisión documental mediante la búsqueda en bases de datos científicas, cuyo acceso es proporcionado por la biblioteca virtual de la UTN. El propósito de la investigación es recolectar información sobre investigaciones científicas que hayan desarrollado servicios de gestión académica, además de conocer sobre el desarrollo de microservicios con NestJS y aplicaciones móviles con Flutter.

2. Para el desarrollo del proyecto se trabajará con la metodología ágil SCRUM, la cual permite la resolución de problemas a corto plazo, reduce los riesgos del proyecto, y ofrece productos o servicios de calidad [4]. Esta metodología se divide en tres fases: Pre-Juego, Juego y Post-Juego. Durante el Pre-Juego, se definirán requisitos. En la fase de Juego, se realizará el desarrollo. Finalmente, en el Post-Juego se llevarán a cabo pruebas de aceptación [5].

3. Una vez finalizado el desarrollo del servicio, se evaluará la calidad de uso mediante la ISO/IEC 25022, dicha norma permite evaluar la calidad de uso de un producto de

software mediante características como la eficacia, eficiencia y satisfacción [6]. Por ende, se llevará a cabo una encuesta a los docentes para conocer el nivel de aceptación de la aplicación.

Justificación

El presente trabajo de titulación se alinea con el objetivo #9 de los Objetivos de Desarrollo Sostenible (ODS): Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación. Se enfoca más específicamente en la meta #9.1, que busca desarrollar infraestructuras digitales confiables y de alta calidad [7]. Buscando así que todas las personas puedan acceder en igualdad de condiciones, sin importar su ubicación o situación económica.

Justificación Tecnológica

Este trabajo busca promover el avance en el desarrollo de arquitecturas con microservicios para aplicaciones móviles. La cual, permite una mayor modularidad y escalabilidad en el desarrollo de aplicaciones móviles, lo que resulta fundamental ante las necesidades y demandas del entorno educativo actual [8]. En relación con los frameworks, el uso de NestJS en el backend permite una arquitectura escalable y modularizada [9], mientras que Flutter en el frontend permite desarrollar aplicaciones móviles altamente eficientes y multiplataforma, destacándose por su capacidad para diseñar aplicaciones intuitivas [10].

Justificación Social

La implementación de aplicaciones móviles es fundamental debido a que no solo pueden satisfacer las necesidades funcionales de los docentes, sino que también garantiza que la tecnología sea fácil de usar y respalde continuamente las expectativas de estos, fomentando la adopción a largo plazo [11]. El servicio de “Portafolio Docente” en la aplicación móvil de la UTN permitirá mejorar y agilizar la gestión académica, fortaleciendo así la comunicación entre docentes, estudiantes y la institución.

Contexto de la Investigación

Tabla 1 Antecedentes

INVESTIGACIÓN	APORTE
Contexto: Local	La tesis consiste en el desarrollo de la Fase 1 del portafolio estudiante de la UTN,

Implementación de portafolio estudiante en la plataforma móvil de la UTN utilizando SCRUM y la norma ISO/IEC 25022 [12].	trabajo que sigue la misma estructura y arquitectura tecnológica que la Fase 1 del servicio de “Portafolio Docente”.
Contexto: Local Desarrollo de una aplicación móvil utilizando el framework Flutter para fomentar el área turística del GAD de Pedro Moncayo [13].	Este trabajo de tesis se enfoca en el desarrollo de una aplicación móvil con Flutter para impulsar el turismo en el cantón Pedro Moncayo, abordando la insuficiente promoción de sus atractivos y eventos.
Contexto: Internacional Developing back-end of a web application with NestJS framework: Case: Integrify Oy's student management system [9].	El trabajo de tesis mencionado explica el desarrollo de un sistema web para la gestión de estudiantes utilizando NestJS en el backend. También se explican tanto las ventajas como desventajas del uso de este framework.
Contexto: Internacional Containerized Microservices for Mobile Applications Deployed on Cloud Systems [14].	El estudio analiza la importancia y la eficiencia del uso de microservicios contenerizados en el desarrollo de aplicaciones móviles en ecosistemas en la nube, destacando cómo el uso de contenedores como Docker o Kubernetes mejora el rendimiento y confiabilidad de la aplicación.
Contexto: Nacional Automation of the Municipal Inspection Process in Ecuador Applying Mobile-D for Android [15].	Se implementó una aplicación móvil en este estudio a través de la metodología Mobile-D con el fin de automatizar las inspecciones manuales en las tiendas por parte del Municipio de Ibarra, Ecuador.
Contexto: Internacional Implementation of Flutter-based Learning Management System (LMS) at Universitas Andi Djemma Palopo [16].	El estudio consiste en la implementación de una aplicación móvil de sistema de gestión del aprendizaje (LMS) desarrollada con Flutter y Django para una universidad. Se implementaron funcionalidades como la gestión de clases, materias, foros, actividades y asistencia.

I MARCO TEÓRICO

El presente capítulo establece los fundamentos teóricos y tecnológicos que sustentan el desarrollo de la Fase 2 del portafolio docente en la aplicación móvil institucional UTN-Móvil. Se abordan los conceptos esenciales relacionados con la arquitectura del sistema, las tecnologías de desarrollo empleadas, la metodología de trabajo adoptada y los estándares de calidad utilizados para la evaluación del producto final. Este marco conceptual proporciona las bases necesarias para comprender las decisiones técnicas y metodológicas aplicadas durante la implementación del proyecto, así como el contexto de trabajos previos y relacionados que han contribuido al desarrollo de soluciones similares en entornos educativos.

1.1 Sistema informático integrado universitario (SIIU)

El Sistema Informático Integrado Universitario (SIIU) de la UTN, también llamado Portafolio Universitario, comenzó en 2007 como una aplicación independiente para la gestión académica. Buscando optimizar las prácticas tanto académicas como administrativas, la universidad adoptó Oracle APEX como su entorno de desarrollo en octubre de 2013. Esta decisión facilitó la automatización de múltiples procesos para los miembros de la comunidad universitaria y permitió la creación de una arquitectura tecnológica unificada [17].

1.1.1 Portafolio Docente UTN

El reglamento establecido del Sistema Informático Integrado Universitario (SIIU) de la Universidad Técnica del Norte establece ocho portafolios clave, siendo el portafolio docente esencial para la administración y el seguimiento de las actividades académicas del profesorado. Con el transcurso del tiempo, este portafolio ha recibido importantes actualizaciones para mejorar la supervisión académica en la UTN. Dos de sus módulos más importantes son la planificación docente y el registro académico, los cuales son fundamentales para el rendimiento de los estudiantes [18].

1.1.2 Implementación previa: funcionalidades desarrolladas en la Fase 1

En la Fase 1 de la implementación del servicio “Portafolio Docente” en UTN-Móvil, se implementaron las siguientes funcionalidades clave: un calendario general para seguir actividades y su estado de calificación; un horario para consultar clases y actividades complementarias; un distributivo que detalla las horas asignadas; un perfil docente con

información profesional centralizada; el registro de asistencia de los estudiantes matriculados; la gestión de avance en las asignaturas; listas detalladas de estudiantes por materia; la opción de descargar reportes de notas del aula virtual y ver el detalle de nota por alumno; el registro académico que lista las materias con su estado diario de asistencia y avance, y permite descargar el sílabo; gestión de actividades para crearlas, editarlas y calificarlas fácilmente; un sistema para enviar notas de parciales y finales, y consultar detalles; y acceso a recursos académicos descargables [3].

1.1.3 Infraestructura tecnológica del SIIU

El SIIU está compuesto de diferentes servicios o portafolios, entre ellos el portafolio estudiante, docente, académico entre otros. Los múltiples servicios fueron desarrollados en Oracle APEX mientras que todos los datos de gestión universitaria como datos de estudiantes, docentes, matrículas, etc. están almacenados en una base de datos de Oracle.

1.1.4 Oracle APEX

Oracle APEX es una plataforma empresarial de bajo código la cual posibilita crear aplicaciones web y móviles seguras y escalables, siendo posible desplegarlas ya sea en la nube o localmente. Se destaca por facilitar el desarrollo rápido e intuitivo de soluciones robustas de software [19]. De esta manera, Oracle APEX reduce la complejidad inherente de desarrollar software y permite desplegar aplicaciones empresariales de todo tipo de industrias a través de una plataforma de bajo costo y con mínimo código. Mediante el *Application Builder*, se pueden desarrollar aplicaciones portables y robustas en el navegador con un rendimiento y experiencia de usuario notables [20].

La fácil integración con las bases de datos Oracle facilita que las aplicaciones Oracle APEX puedan manejar de manera eficiente el acceso y manipulación de las cantidades de datos robustas que almacenan dichas bases de datos, al procesar volúmenes de datos enormes y garantizando un alto rendimiento [21].

1.1.5 Base de datos Oracle

Oracle Database es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) diseñado para almacenar, gestionar y recuperar de forma eficiente y segura grandes cantidades de datos en entornos multiusuario [22]. Siendo uno de los sistemas de gestión de bases de datos más populares, este éxito se debe en gran medida a medida a su alta

conurrencia y baja latencia, lo cual permite a los usuarios registrar y procesar todo tipo de solicitudes o transacciones de clientes u otro rol dentro un negocio [23]. Su arquitectura se basa en una base de datos física y al menos una instancia que la gestiona. La interacción con los datos, organizados en esquemas con objetos como tablas e índices, se realiza principalmente a través de SQL, con extensiones procedurales como PL/SQL y soporte para Java para la lógica del servidor. Se caracteriza por su robusta gestión de transacciones, control de concurrencia y garantía de consistencia de los datos [22].

1.2 Arquitectura de microservicios en el backend

Los sistemas de software monolíticos tienden a ser considerados cada vez más anticuados y antiguos debido principalmente a sus limitaciones relacionadas a la capacidad de crecimiento, facilidad de mantenimiento, operatividad continua y eficiencia. Esto subraya la urgencia de modernizarlos para potenciar su rendimiento y ofrecer una mejor experiencia integral a quienes los desarrollan y utilizan [24].

Ante esta necesidad de modernizar los sistemas de software, surge la arquitectura de microservicios. En una arquitectura de microservicios se dividen las funciones de una aplicación en varios servicios más pequeños que están conectados entre sí. La ventaja de este enfoque es que cada uno de estos pequeños servicios se puede implementar o modificar por separado y cuantas veces sea necesario, sin afectar el funcionamiento general del conjunto de servicios y asegurando que todo el sistema siga operando correctamente [25].

1.2.1 Principios de la arquitectura de microservicios

Para entender cómo el paradigma de los microservicios logra sus objetivos, es esencial conocer las características o principios fundamentales que rigen su diseño e implementación.

Autonomía: Los microservicios se pueden desarrollar, implementar, operar y escalar por separado, sin afectar cómo funcionan los demás servicios, debido a que, cada uno microservicio funciona como una unidad independiente. Además, no necesitan compartir ni estructura ni código interno. Y para la comunicación entre ambos, se realiza a través de interfaces bien definidas como lo son APIs [26].

Especialización: A cada microservicio se le asigna un conjunto de responsabilidades o tareas específicas, de modo que, estos se enfocan en resolver problemas particulares. Si a lo largo del ciclo de vida del software, un servicio empieza a volverse demasiado robusto y complejo

para su mantenimiento, dicho servicio puede dividirse en servicios más especializados y pequeños, facilitando el trabajo de los desarrolladores [26].

Descentralización: La separación de funcionalidades y del flujo de los datos en microservicios especializados mejora la privacidad de la información. De tal manera que, si se explota una vulnerabilidad en uno de los microservicios, el alcance del ataque se limita potencialmente a solo unos pocos componentes. Y en tal situación, revisar y solucionar las vulnerabilidades es más fácil al tratarse de un número reducido de componentes [27].

1.2.2 Comunicación entre microservicios y APIs RESTful

La comunicación entre microservicios es un pilar fundamental en las arquitecturas modernas de software, ya que permite la interoperabilidad y la escalabilidad de sistemas distribuidos. Las APIs RESTful (Representational State Transfer) son el estándar predominante para facilitar esta comunicación debido a su simplicidad, flexibilidad y compatibilidad con los principios de la web. REST utiliza los métodos HTTP estándar (GET, POST, PUT, DELETE, etc.) para realizar operaciones sobre recursos identificados por URIs, promoviendo un diseño *stateless* y orientado a recursos [28].

Las arquitecturas de microservicios suelen gestionar su comunicación mediante dos vías fundamentales. Para el intercambio de mensajes dentro del propio sistema, entre sus diferentes componentes de servicio, se opta por un enfoque basado en eventos, como es el caso del patrón publicación-suscripción. En cambio, cuando se necesita interactuar con servicios fuera de la arquitectura, se recurre a un esquema de petición y respuesta, del cual REST (Transferencia de Estado Representacional) es un ejemplo representativo [29].

1.2.3 Lenguaje de programación TypeScript

Pese a su popularidad y amplio uso en el desarrollo web a lo largo de los años, manteniéndose como el lenguaje de programación más usado, de acuerdo con la encuesta de Stack Overflow de 2024 [30], JavaScript es criticado por ciertas limitaciones importantes para el desarrollo y mantenimiento de aplicaciones de software de gran escala. Ante la necesidad emergente de solucionar las deficiencias del JavaScript, surge TypeScript como una extensión y alternativa de JavaScript, para ser un verificador de tipos estático para aplicaciones JavaScript; por lo que, TypeScript es una herramienta que verifica que los tipos de la aplicación sean

correctos, aplicando dicha verificación antes de la ejecución del código, de tal manera que se detectan errores antes de correr el programa [31].

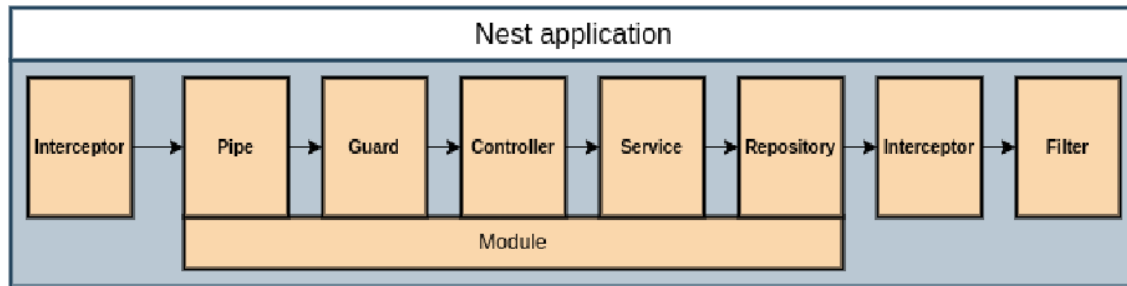
TypeScript es un lenguaje de código abierto desarrollado y mantenido por Microsoft. Tiene dos componentes principales: el código y la definición de tipos. El código en sí es JavaScript con la sintaxis adicional específica de TypeScript, por lo que un programa TypeScript es capaz de correr en cualquier entorno compatible con JavaScript, ya que, al compilarse, todas las partes de TypeScript son eliminadas dejando únicamente código JavaScript. Por otra parte, las definiciones de tipo hacen referencia a la estructura del código JavaScript existente. Los archivos de TypeScript (.d.ts) no contienen una implementación real, sino que describen los tipos, lo que permite garantizar la interoperabilidad con JavaScript mencionada previamente [32]. Además, TypeScript mejora JavaScript al incorporar un sistema de módulos, clases, interfaces y un sistema de tipos estáticos. Con el objetivo de ofrecer una ayuda práctica y ligera a los desarrolladores, tanto el sistema de módulos como el de tipos son flexibles y de fácil adopción [33].

1.2.4 Framework NestJS para desarrollo de microservicios

NestJS es un framework de desarrollo backend diseñado para para construir aplicaciones escalables y eficientes con Node.js de forma organizada. Utiliza el lenguaje de programación Typescript (aunque soporta también JavaScript) y se fundamenta en conceptos de programación orientada a objetos, funcional y reactiva. NestJS está basado en Express (o Fastify) y se inspira en la arquitectura framework Angular. Entre sus principales características destaca su enfoque de módulos, la inyección de dependencias, la documentación detallada, la integración con múltiples bases de datos, la facilidad para realizar pruebas, y una interfaz de comandos para analizar y generar código, entre otras [34].

Una aplicación NestJS generalmente tiene la siguiente arquitectura, en donde los principales componentes son los siguientes:

Figura 3 Arquitectura de una aplicación Nest



Fuente: [35]

- **Módulos**

Todas las aplicaciones Nest se organizan en módulos, al menos cada una tiene un módulo raíz, el cual actúa como punto inicial de la aplicación para construir el grafo de la aplicación, dicho grafo es una estructura interna que permite resolver dependencias y relaciones entre los diferentes módulos o *providers* dentro de una aplicación Nest. En aplicaciones grandes, como la aplicación backend del servicio “Portafolio Docente” implementada en la fase 1, se organizan en múltiples módulos, cada uno representado un componente clave de la aplicación, en donde se encapsulan un conjunto de capacidades [36].

- **Providers**

Siendo un componente clave de Nest, los *providers* consisten en clases como *services*, *repositories*, *factories* y *helpers*, que pueden ser inyectadas como una dependencia. Esto permite a los objetos establecer múltiples relaciones entre ellos. El sistema de ejecución de Nest se encarga automáticamente de conectar estos objetos y dependencias entre sí [37].

- **Pipes**

Los *pipes* son clases inyectables que permiten validar y transformar los datos entrantes de una solicitud, por ejemplo, convertir un valor de tipo cadena a entero. Estos actúan en el *controller*, justo antes de que invocar al método correspondiente. NestJS incluye varios pipes integrados por defecto, aunque también se pueden crear *pipes* personalizados para aplicar lógica de negocio propia según los requisitos de la aplicación [38].

- **Controllers**

Los *controllers* son otro componente esencial en Nest, así como en cualquier aplicación web, ya que se encargan de administrar las peticiones HTTP enviadas al servidor. El mecanismo de ruteo decide cuál *controller* es el encargado de manejar una solicitud específica. Por lo general, un *controller* tiene múltiples rutas, mientras que cada ruta realiza una acción específica [39].

1.3 Aplicaciones móviles

Las aplicaciones móviles hacen referencia a programas de software que pueden ejecutarse en dispositivos móviles o inteligentes, típicamente disponibles en tiendas de aplicaciones como lo son Google Play o Apple App Store [40]. Representan un sector en rápido crecimiento dentro de las tecnologías de la información y comunicación a nivel global. Estas aplicaciones se caracterizan por ser fáciles de usar, amigables con el usuario, económicas y compatibles con la mayoría de los teléfonos móviles, incluso los de gama baja o más básicos [41]. Pese a compartir similitudes técnicas con otros tipo de software, como las aplicaciones web, las aplicaciones móviles se diferencian en otras características o factores a tener cuenta en el desarrollo de aplicaciones: manejo de sensores, como lo son sus cámaras integradas y pantallas táctiles; la gran variedad de dispositivos y plataformas de hardware; consumo de energía; o el tamaño generalmente pequeño de los dispositivos móviles [40].

1.3.1 Aplicaciones móviles en entornos universitarios

Las aplicaciones móviles han revolucionado la forma en que los estudiantes y docentes interactúan con el conocimiento en el contexto universitario. El crecimiento exponencial en el uso de dispositivos móviles ha impulsado el desarrollo de millones de aplicaciones de software para estos dispositivos móviles, comúnmente denominadas 'apps' [42]. Este fenómeno ha llevado a las instituciones de educación superior a adoptar estrategias tecnológicas que aprovechan las capacidades de los dispositivos móviles para mejorar los procesos de enseñanza y aprendizaje.

Las aplicaciones móviles tienen la capacidad de desempeñar un rol fundamental en la educación, proporcionando ventajas al proceso de aprendizaje y enseñanza, entre ellas: aprendizaje interactivo, mayor accesibilidad a recursos educativos y personalización del aprendizaje. Por ende, los estudiantes pueden satisfacer sus necesidades e intereses educativos al acceder y participar con los recursos de la institución educativa desde cualquier lugar. Pese a las ventajas mencionadas, existen ciertas desventajas inherentes al uso de aplicaciones móviles

educativas, como lo son: fácil distracción de los usuarios, dependencia excesiva a la tecnología y acceso no equitativo para quienes no poseen un dispositivo móvil [43].

Asimismo, las aplicaciones móviles han mejorado la comunicación entre docentes y estudiantes, permitiendo una interacción más fluida y eficiente. El desarrollo de aplicaciones específicas para este propósito ha demostrado ser una solución efectiva para superar las barreras de comunicación en los entornos universitarios [44].

1.3.2 Tendencias del uso de plataformas móviles

Según el Unión Internacional de Telecomunicaciones, o ITU por sus siglas en inglés, un 80% de la población mundial tiene un dispositivo móvil y un 68% posee también acceso a internet [41]. Por ende, las tendencias actuales en el uso de plataformas móviles muestran una evolución significativa hacia la integración de tecnologías emergentes y enfoques centrados en el usuario.

Debido al progreso en la tecnología y la infraestructura móvil, los teléfonos móviles se han vuelto indispensables en la vida cotidiana actual. Esta situación ha generado una notable necesidad de software y aplicaciones de un nivel superior en este ámbito [45].

Esta evolución ha generado nuevos desafíos y oportunidades en el desarrollo de aplicaciones móviles, particularmente en el contexto educativo, donde la necesidad de herramientas especializadas y la demanda de experiencias de aprendizaje personalizadas continúan impulsando la innovación tecnológica.

1.3.3 Tipos de aplicaciones móviles

Las aplicaciones móviles se desarrollan principalmente de tres maneras: como aplicaciones nativas, web o híbridas. El enfoque nativo permite a los desarrolladores aprovechar al máximo todas las capacidades del dispositivo, resultando en un mejor rendimiento. Sin embargo, crear una aplicación nativa para múltiples plataformas exige un conocimiento especializado de los lenguajes de programación, las APIs y kits de desarrollo de software (SDK) específicos de cada una.

Por otra parte, las aplicaciones móviles web son accesibles desde navegadores móviles, debido a que se ejecutan en servidores web que son independientes al dispositivo móvil, destacando su alta portabilidad en diferentes plataformas, aunque careciendo de funcionalidades relacionadas con el hardware del dispositivo, como la cámara o acelerómetro [46].

Por último, las aplicaciones híbridas se consideran aplicaciones web que son empaquetadas y compiladas para funcionar y actuar como nativas, por lo que están disponibles tanto en Android como en iOS. Estas aplicaciones realizan llamadas a APIs para obtener datos y mostrarlos dentro de la interfaz de usuario. Debido a la alta dependencia a servicios de internet, las aplicaciones híbridas generalmente tienen un rendimiento más lento a comparación con las nativas [47].

1.3.4 Aplicaciones móviles multiplataforma

En los últimos años, uno de los avances más destacados en tecnología ha sido el crecimiento acelerado de la computación móvil y la popularidad de las aplicaciones para dispositivos móviles. A medida que estos dispositivos mejoran en capacidad, las aplicaciones también se vuelven más complejas y sofisticadas. Sin embargo, estas aplicaciones enfrentan desafíos particulares como la necesidad de ser rápidas, consumir poca memoria y poca energía. Además, existen diversas plataformas móviles en competencia, principalmente Android y iOS, lo que complica el desarrollo, ya que cada una utiliza diferentes lenguajes de programación y APIs. Esto hace que adaptar una app de una plataforma a otra sea costoso y demore tiempo. Actualmente, existen varias soluciones para crear aplicaciones móviles que funcionen en múltiples plataformas, aunque cada una presenta diferencias en tecnología, costos, rendimiento y limitaciones [48].

Las aplicaciones móviles multiplataforma han ganado relevancia en el desarrollo de software debido a su capacidad para operar en múltiples sistemas operativos utilizando una única base de código. Este enfoque contrasta con las aplicaciones nativas, que requieren desarrollos específicos para cada plataforma, como iOS o Android, utilizando lenguajes y herramientas particulares.

1.3.5 Flutter como framework multiplataforma

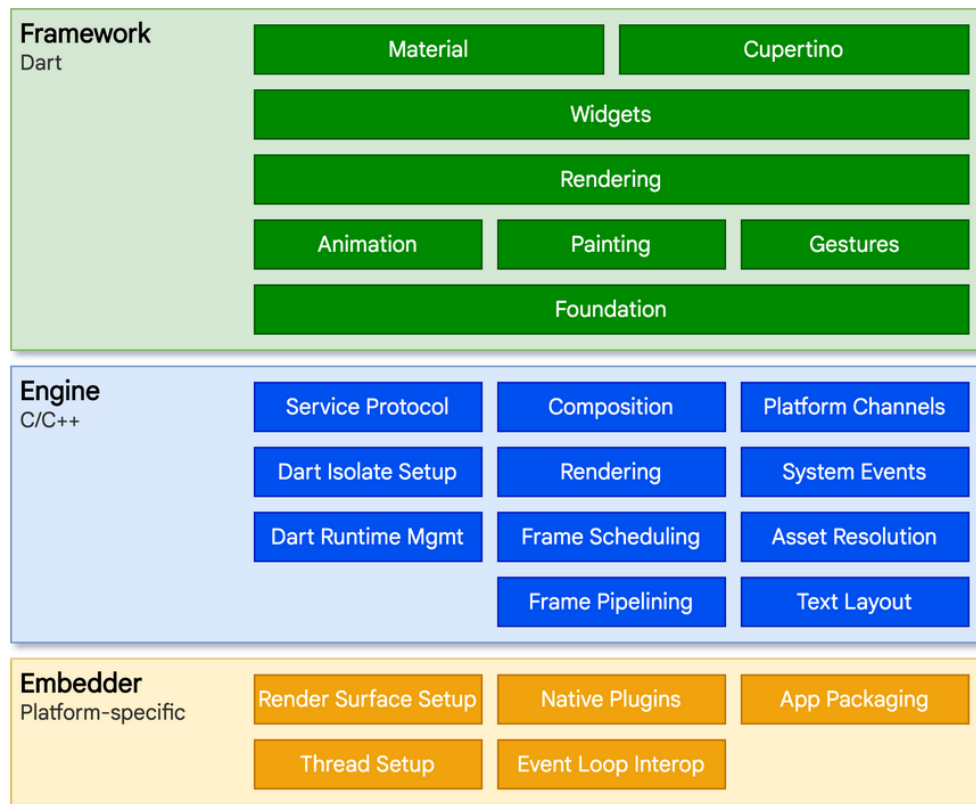
Considerando el desafío y la inversión de recursos que implica el desarrollo nativo para múltiples plataformas [46], es pertinente explorar herramientas y frameworks diseñados para optimizar este proceso y ofrecer alternativas eficientes. En este contexto, surge Flutter como una solución moderna que propone un enfoque unificado para la creación de interfaces de usuario atractivas y compiladas de forma nativa para diversas plataformas. Flutter es un framework de código abierto que permite crear aplicaciones atractivas para múltiples plataformas, las cuales se compilan de forma nativa a partir de una única base de código, fomentando la reutilización de

este. Este framework posibilita la creación de aplicaciones para dispositivos móviles, web, de escritorio y embebidos, en plataformas como Android, iOS y escritorio, todo esto a partir de una única base de código [49].

Arquitectura de Flutter

Flutter tiene una arquitectura basada en capas compuesta por diferentes librerías independientes. La cual se representa en la siguiente figura.

Figura 4 Arquitectura de una aplicación Flutter



Fuente: [50]

Las aplicaciones Flutter son construidas como cualquier aplicación nativa. El punto de partida es el *embedder*, específico de cada plataforma, que coordina el acceso a servicios del sistema operativo subyacente, como renderizado de superficies y entradas, además de gestionar el bucle de eventos de mensajes. El motor de Flutter es el encargado de pintar nuevos fotogramas mediante la rasterización de escenas compuestas, también proporciona una implementación de bajo nivel de la API principal de Flutter, en donde se incluyen gráficos, E/S de archivos y red, soporte de accesibilidad, herramientas de ejecución de Dart, entre otros. Por

último, el framework Flutter es responsable de proporcionar clases fundacionales, la capa de renderizado de objetos, la capa de *widgets* que permite la programación reactiva y librerías *Material* y *Cupertino* que gestionan controles asociados con *widgets* [50].

Widgets

Flutter tiene un enfoque de desarrollo basado en *widgets*, en donde los mismos se pueden considerar como bloques de construcción, los cuales pueden ser utilizados y reusados para construir las interfaces de usuario [51]. Todo se considera como un *widget* dentro de Flutter, cada uno permite describir todas las características de una interfaz de usuario, incluyendo aspectos físicos, como botones y texto, y de diseño, como el margen o padding [52].

Flutter denomina a los *widgets* como unidades de composición, en las que cada uno forma una jerarquía basada en la composición. De esta forma, cada *widget* tiene una conexión con su padre y recibe su contexto. Esta estructura llega hasta al *widget* raíz, que representa el contenedor que aloja la aplicación Flutter, típicamente *MaterialApp* o *CupertinoApp* [52]. En base a las interacciones del usuario, el framework actualiza la aplicación al reemplazar un *widget* con otro dentro de la jerarquía, de tal manera que se realiza la actualización de forma eficiente al comparar nuevos y viejos *widgets* [50].

1.3.6 Lenguaje de programación Dart

Dart se caracteriza por ser un lenguaje de programación optimizado para desarrollar aplicaciones eficientes en cualquier plataforma, con el fin de ser el lenguaje multiplataforma más competente y productivo. Mediante las herramientas técnicas proporcionadas, como el *hot reload*, Dart permite el desarrollo de aplicaciones con alta calidad en producción a lo largo de diferentes plataformas como escritorio, móvil y web. Además, Dart se complementa con Flutter al ser la base fundamental del mismo, en donde Dart se encarga de proporcionar el lenguaje y tiempos de ejecución. Aunque es un lenguaje de tipado estático, Dart también es flexible al permitir el uso de variables dinámicas si es necesario. A esta flexibilidad, se suma la característica de que Dart tiene un mecanismo para asegurar que ningún valor nunca sea nulo, por lo que se evitan excepciones relacionadas a valores nulos en tiempo de ejecución [53].

Además de las características mencionadas, Dart también ofrece la ventaja clave de garantizar un alto rendimiento en producción. Esto se consigue, debido a que, Dart compila el código fuente en código de máquina nativo mediante compilación anticipada (AOT), de tal manera que se optimiza la utilización de recursos y tiempos de ejecución [51].

1.4 Metodología ágil de desarrollo

En el desarrollo de software existen dos enfoques principales que se han establecido a lo largo de los años: las metodologías tradicionales impulsadas por un plan y las metodologías ágiles. Una metodología tradicional basada en la planificación se caracteriza por una planificación exhaustiva, una orientación a procesos, documentación detallada y un enfoque predictivo. Por otro lado, la metodología ágil fomenta la entrega frecuente de incrementos de software en ciclos de iteración cortos y promueve el trabajo en equipo, la colaboración con el cliente y un enfoque adaptable [54].

Según Beck et al. los principios ágiles se fundamentan en cuatro valores principales: individuos e interacciones sobre procesos y herramientas, software funcionando sobre documentación exhaustiva, colaboración con el cliente sobre negociación contractual, y respuesta ante el cambio sobre seguir un plan [55].

1.4.1 Fundamentos de metodologías ágiles y marco Scrum

Scrum es una de las metodologías ágiles de desarrollo más ampliamente utilizadas en diferentes industrias, lo que implica su significativa importancia en el desarrollo moderno de software [56]. Scrum fue propuesto por Ken Schwaber y Jeff Sutherland para abordar los desafíos del desarrollo de productos complejos [57]. Schwaber y Sutherland definen Scrum como un marco ligero que facilita que personas, grupos y compañías desarrollen soluciones adaptativas que generen valor al abordar situaciones difíciles. [58]. El marco se basa en tres pilares fundamentales: transparencia, inspección y adaptación, que permiten la mejora continua del proceso de desarrollo.

1.4.2 Roles

El marco Scrum define tres roles fundamentales que distribuyen responsabilidades específicas para asegurar el éxito del equipo:

Product Owner: es la figura única y fundamental dentro del equipo, cuya principal misión es maximizar el valor del producto que se desarrolla. Es el responsable final de la gestión efectiva del Product Backlog, lo que implica crear, comunicar claramente y priorizar los elementos de trabajo (ítems), así como definir y difundir el objetivo general del producto (Product Goal), asegurándose de que el backlog sea transparente, visible y entendido por todos [58].

Scrum Master: es la persona responsable de implantar Scrum tal como se describe en la Guía Scrum, ayudando a que tanto el equipo Scrum como la organización en general comprendan su teoría y práctica. También se encarga de la efectividad del equipo Scrum, facilitándoles la mejora de sus procesos dentro del marco de trabajo Scrum. En esencia, los Scrum Masters son verdaderos líderes que están al servicio del equipo Scrum y de la organización en un sentido más amplio [58].

Equipo de Desarrollo: son aquellos integrantes comprometidos con la creación de cualquier faceta de un Incremento que sea utilizable al final de cada Sprint. Los desarrolladores siempre tienen la responsabilidad de elaborar el Sprint Backlog, de asegurar la calidad adhiriéndose a una Definición de Hecho, de ajustar su plan cada día con la mira en el Sprint Goal y de responsabilizarse mutuamente como profesionales en su labor [58].

1.4.3 Artefactos

Los artefactos de Scrum representan trabajo o valor para proporcionar transparencia y oportunidades para la inspección y adaptación [59]. Los tres artefactos principales son:

Product Backlog: Lista dinámica y organizada de todos los requisitos necesarios para un proyecto Scrum, originada a partir de la visión del producto definida por el Product Owner. Esta lista se organiza según el valor que cada elemento aporta al cliente o al negocio, situando los ítems más valiosos en la parte superior. A lo largo de la vida del proyecto, el Product Backlog está en constante evolución, ya que continuamente se le añaden, eliminan o se vuelve a priorizar su contenido [57].

Sprint Backlog: El Sprint Backlog se crea al inicio de cada Sprint durante la reunión de planificación, donde el Product Owner y el equipo Scrum, con la ayuda del Scrum Master, revisan el Product Backlog general. En esta reunión, el equipo Scrum selecciona un conjunto de elementos del Product Backlog, comenzando por los de mayor prioridad, a los que se comprometen a completar para el final del Sprint. Posteriormente, cada uno de estos elementos seleccionados se diseña y se desglosa en una serie de tareas individuales [57].

Sprint: el desarrollo de productos se organiza en ciclos de trabajo denominados Sprints, que son periodos o iteraciones laborales cuya duración suele ser de una a cuatro semanas. Estos Sprints se caracterizan por tener una duración fija y concluyen en una fecha preestablecida, sin

importar si todo el trabajo planificado se ha finalizado o no; nunca se prolongan más allá de este límite temporal [57].

1.4.4 Fases de desarrollo: Pre-Juego, Juego y Post-Juego

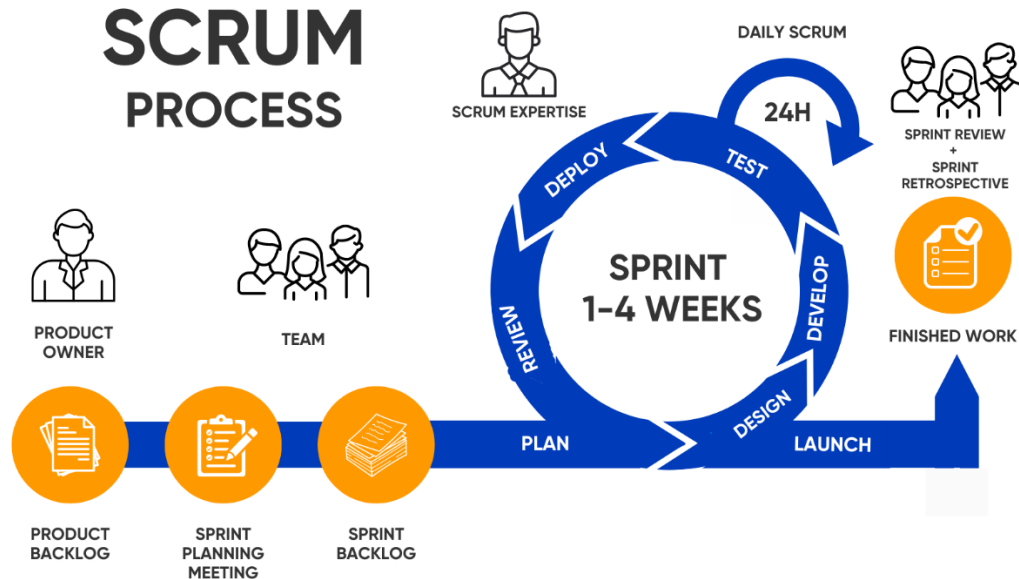
Según la guía metodológica propuesta por Huertas [5], un proyecto Scrum se divide en las siguientes fases de desarrollo:

Pre-Juego: Es la etapa inicial de planificación donde se define la visión del proyecto, se establece el presupuesto, se crea el listado inicial de requisitos (Product Backlog) y se conforma el equipo Scrum con sus roles. Se priorizan tareas y se diseñan los primeros elementos, incluyendo análisis funcionales y técnicos [5].

Juego: Corresponde a la fase de desarrollo activo, donde el equipo trabaja en ciclos iterativos llamados Sprints para construir el producto. Se crea un plan detallado para cada Sprint (Sprint Backlog), se realizan reuniones diarias para coordinar, y al final de cada ciclo se revisa el trabajo completado y se reflexiona sobre el proceso en una retrospectiva [5].

Post-Juego: En esta fase final, se evalúan los incrementos del producto desarrollados con el cliente para asegurar que cumplan con los requisitos. Si todo es aprobado, se procede a la entrega del producto final; de lo contrario, se puede regresar a la fase inicial para realizar ajustes [5].

Figura 5 Diagrama del proceso Scrum



Fuente: [60]

1.5 Evaluación de calidad en uso del software

La evaluación de la calidad en uso del software ha adquirido una importancia fundamental en el desarrollo de sistemas de información, ya que permite determinar el grado en que un producto satisface las necesidades específicas de los usuarios finales en contextos reales de uso. Esta evaluación trasciende las métricas técnicas tradicionales para enfocarse en la experiencia del usuario y la efectividad del sistema para alcanzar objetivos específicos, convirtiéndose en un factor crítico para el éxito de cualquier producto de software [61].

1.5.1 Norma ISO/IEC 25010: Modelo de calidad del producto de software

La norma ISO/IEC 25010 forma parte de la familia de estándares SQuaRE (System and Software Quality Requirements and Evaluation) de la serie ISO 25000, y establece un modelo de calidad para productos de software y sistemas [62]. Esta norma, que reemplazó al modelo ISO/IEC 9126, define un marco comprehensivo para la evaluación de la calidad del software mediante la especificación de características y subcaracterísticas de calidad.

Figura 6 Divisiones de la familia de normas ISO/IEC 25000



Fuente: [63]

1.5.2 Calidad en uso según la norma ISO/IEC 25010.

La calidad en uso representa el grado en que un producto utilizado por usuarios específicos satisface sus necesidades para alcanzar objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso específico. La ISO/IEC 25010 define cinco características principales de calidad en uso que proporcionan un marco integral para evaluar la experiencia del usuario final [61].

1.5.3 Norma ISO/IEC 25022: Medición de la calidad en uso

La ISO/IEC 25022 establece medidas específicas para evaluar la calidad en uso, que es una perspectiva centrada en el usuario final. Estas medidas permiten determinar en qué medida un producto o sistema permite a los usuarios alcanzar sus objetivos de manera efectiva, eficiente y satisfactoria en un contexto específico de uso [64].

La norma se utiliza en conjunto con otros estándares de la serie ISO/IEC 25000, como la ISO/IEC 25010, que define el modelo de calidad, y la ISO/IEC 25021, que especifica elementos de medida de calidad. Esta integración facilita una evaluación completa y coherente de la calidad del software desde múltiples perspectivas [64].

1.5.4 Escalas e instrumentos de medición de calidad en uso

Considerando que la presente tesis es la Fase 2 de la implementación del portafolio docente, se evaluarán las mismas características y sub características elegidas en la Fase 1, de tal manera que se puedan comparar los resultados de ambos trabajos de manera más directa y sencilla [3]. Cada una de estas características se define y se mide de la siguiente manera:

Eficacia: Se refiere al grado en que los usuarios pueden alcanzar sus objetivos específicos con precisión y completitud en un contexto de uso particular. Las métricas asociadas incluyen el porcentaje de tareas completadas con éxito y la tasa de errores cometidos durante la realización de tareas [64].

Eficiencia: Mide los recursos utilizados en relación con la precisión y completitud con la que los usuarios alcanzan sus objetivos. Esto puede incluir el tiempo requerido para completar tareas y la cantidad de esfuerzo necesario [64].

Satisfacción: Evalúa el grado en que las necesidades y expectativas de los usuarios se cumplen al utilizar el sistema. Las métricas pueden incluir encuestas de satisfacción del usuario y análisis de retroalimentación cualitativa [64].

La norma ISO/IEC 25022 proporciona un marco estructurado para medir la calidad en uso de sistemas de software, enfocándose en cómo los usuarios interactúan con el sistema en su entorno real. Al utilizar métricas específicas para eficacia, eficiencia y satisfacción, las organizaciones pueden identificar áreas de mejora y asegurar que los sistemas desarrollados cumplan con las necesidades y expectativas de los usuarios finales.

1.6 Trabajos relacionados

Considerando que el presente trabajo de titulación corresponde a la Fase 2 de la implementación del servicio “Portafolio Docente”, resulta evidente que el trabajo relacionado más importante es la Fase 1 del mismo. Enríquez [3] implementó la primera fase mediante el desarrollo del servicio en la aplicación UTN Móvil para así mejorar la gestión académica docente en dispositivos móviles al proporcionar una aplicación móvil dedicada y optimizada. Se estableció una arquitectura de microservicios en el backend mediante NestJS, la cual se comunica con la aplicación móvil desarrollada en Flutter, durante el desarrollo se empleó la metodología ágil SCRUM. La evaluación de la calidad en uso, basada en la norma ISO/IEC 25022 y aplicada a 22 docentes de todas las facultades de UTN, presentó un resultado del 87.63%, demostrando una

implementación satisfactoria del servicio que mejora y fortalece la gestión académica docente. Por limitaciones de tiempo se descartó adaptar funcionalidades clave de la plataforma web a la aplicación móvil por lo que la misma no cubre todas las necesidades de los docentes, además de que los docentes realizaron muchas observaciones demostrando una confianza media del 50%.

Oliveira y Matos [65] implementaron una aplicación móvil para adaptar a dispositivos móviles el *Sistema de gestión del aprendizaje* (LMS) utilizado en su universidad, el cual está orientado principalmente a su uso en computadoras de escritorio, mejorando y optimizando así la gestión académica y docente. Al tratarse de la adopción de Sakai LMS a móviles, la aplicación desarrollada en React Native centraliza el acceso a la información mediante una API Rest expuesta por un servidor intermedio desarrollado en PHP con MySQL, a la vez que hace uso de los servicios web existentes de Sakai. Como resultado se desarrolló una API eficiente que se complementa con los servicios de Sakai y permitió la implementación de una aplicación móvil útil e intuitiva para estudiantes y docentes. La aplicación tiene una dependencia excesiva al *Web Scrapping* realizado a la plataforma Sakai LMS y, además, muestra deficiencias en la sincronización automática de varios módulos o funcionalidades.

Aldás [66] diseñó e implementó en su trabajo de grado una aplicación móvil educativa multiplataforma para fomentar el aprendizaje de la lengua de señas ecuatoriana y así disminuir la brecha social entre la comunidad sorda y personas oyentes. Mediante la metodología ágil de desarrollo Mobile-D, se construyó la aplicación móvil utilizando el framework frontend Flutter, mientras que en el backend se emplearon los servicios de Firebase, de tal manera que se implementó la gestión de los datos, solicitudes y autenticación. A través de pruebas de funcionalidad aplicadas a estudiantes y administradores, la aplicación desarrollada tuvo resultados positivos, al implementar funcionalidades como el diccionario de señas, el perfil, los recursos de aprendizaje, las lecciones y los juegos. Una limitación considerable de la aplicación móvil es la dependencia total a internet y de los servicios de Firebase, lo que restringe su disponibilidad para personas sin una conexión constante.

Li y Zhang [67] desarrollaron una plataforma educativa online basada en una arquitectura de microservicios para conseguir una gestión dinámica y distribución eficiente de los recursos educativos en línea. La capa de presentación de la plataforma web se desarrolló utilizando el framework frontend Vue.js, por otro lado, la capa de negocios, basada en un entorno de microservicios, se implementó en el backend mediante los frameworks Spring Boot y Spring

Cloud, y, por último, en la capa de persistencia, la gestión de datos se realizó a través de la base de datos relacional MariaDB. Se desplegó una plataforma web de interacción rápida con alta disponibilidad y escalabilidad, ofreciendo funcionalidades como la gestión de usuarios, cursos, capítulos e instructores. En la plataforma web desarrollada, el API Gateway se encarga de muchas funcionalidades claves de seguridad, por lo que la arquitectura de microservicios se beneficiaría de contar con servicios centralizados, como uno de autorización.

En su tesis, Morales [68] diseñó e implementó una aplicación móvil para evaluar el desempeño docente en su universidad, la cual permite a los estudiantes calificar a los docentes al final de cada clase de forma rápida y anónima. Mediante la metodología ágil SCRUM, se construyó la aplicación móvil utilizando el framework frontend Flutter y aplicando el patrón de diseño MVC, por otra parte, las solicitudes y la base de datos se gestionan mediante Supase, que proporciona una base de datos PostgreSQL integrada en un backend completo. Mediante pruebas funcionales, se verificó el correcto funcionamiento de los diferentes módulos de la aplicación, mientras que, a través de pruebas de usabilidad aplicadas a un grupo de 30 estudiantes, la mayoría ratificaron que la aplicación es intuitiva y fácil de usar. Las funcionalidades orientadas al docente se enfocan únicamente en la recolección y visualización de datos, careciendo de herramientas más avanzadas para un análisis profundo o la toma de decisiones.

Aung et al. [69] implementaron una plataforma educativa para apoyar a principiantes en el aprendizaje del framework de desarrollo Flutter. La plataforma integra 4 componentes en contenedores Docker. La plataforma de respuestas, desarrollada con Node.js y EJS, captura el código realizado por el estudiante. Tras esto, el entorno Flutter compila el código Dart enviado. Luego, el servidor web Ngnix renderiza el código JavaScript compilado a partir del código Flutter/Dart y muestra una imagen UI en el navegador de la actividad hecha por el estudiante. Finalmente, la herramienta de pruebas de UI captura y compara las imágenes UI generadas con imágenes correctas mediante algoritmos de OpenCV para verificar la solución. A través de un taller práctico y evaluación SUS aplicada a estudiantes graduados se concluyó la alta efectividad de la plataforma en guiar a estudiantes en el proceso de aprendizaje, aunque presenta problemas de usabilidad, obteniendo un puntaje de 66%. La configuración inicial para instalar la plataforma es compleja para usuarios no familiarizados con Docker, además que, la evaluación realizada tiene un alcance muy limitado aplicado a solo 10 estudiantes, por lo que los resultados no son muy generalizables.

Laswi et al. [16] desarrollaron una aplicación móvil LMS para su universidad, con el fin de incentivar el e-learning y cumplir con las necesidades de docentes y estudiantes. Empleando la metodología ADDIE, se desarrolló la aplicación móvil utilizando el framework frontend Flutter, en la que se incluyeron funcionalidades como: creación de cursos, discusión y foros, creación de materias, tareas y asistencia. Además, para el backend, se integraron los servicios de Firebase, incluyendo la gestión de base de datos y autenticación, destacando la obtención de datos en tiempo real. En base a una evaluación de usabilidad, se determinó la aplicación como muy factible con un porcentaje de elegibilidad obtenido del 83,03%. Al tratarse de un LMS, la propuesta implementada carece de módulos o funcionalidades críticas como evaluaciones, calificación de notas, una gestión más robusta de los cursos y reportes de auditoría más avanzados.

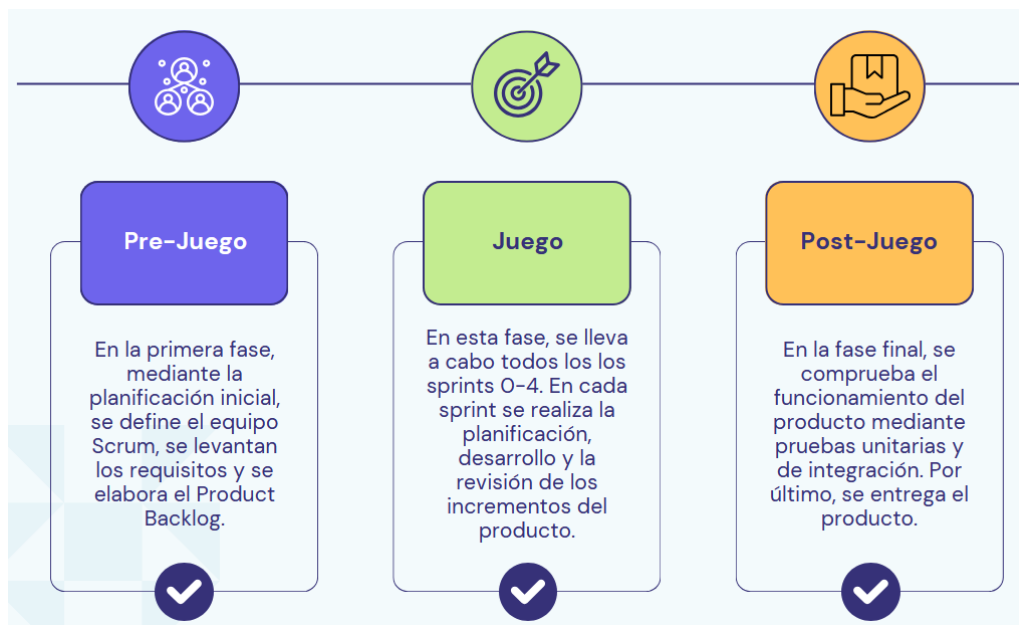
Sulla-Torres et al. [70] implementaron una aplicación móvil enfocada en el cálculo de la densidad mineral ósea(BMD) hacia escolares, con el objetivo de detectar la osteoporosis. Aplicando la metodología Mobile-D, se desarrolló una aplicación móvil mediante Android Studio, utilizando los servicios de Firebase para la autenticación y gestión de base de datos. En ella, los estudiantes ingresan sus atributos antropométricos para calcular la BMD y mediante ecuaciones de regresión antropométrica se determina el grado de osteoporosis. De acuerdo con la ISO/IEC 25022, se evaluó la calidad en uso de la aplicación móvil, obteniendo un 33% de eficacia, un 28% de eficiencia, un 22% de satisfacción, un 5% de libertad de riesgo y un 5% de cobertura del contexto. Estos resultados suman un 93%, correspondiendo a una calificación aceptable. Las ecuaciones de regresión propuestas fueron obtenidas a partir de datos de escolares de Arequipa, Perú, representando una deficiencia crítica en la capacidad de generalización de la aplicación.

Fierro [12] implementó la Fase 1 del portafolio estudiante en la aplicación UTN Móvil para mejorar la experiencia de usuario de los estudiantes en dispositivos móviles. Se adaptaron las funcionalidades del SIU existentes en el nuevo servicio empleando la metodología SCRUM. Basado en una arquitectura de microservicios en el backend, desarrollada con el framework NestJS e integrada con la base de datos Oracle establecida en la universidad, se implementó la aplicación móvil multiplataforma mediante el framework frontend Flutter. Se desarrollaron funcionalidades clave como la gestión de tareas, revisión de notas, calendario de actividades, perfil del estudiante, entre otras. Se realizó una evaluación de la calidad en uso fundamentada en la norma ISO/IEC 25022 aplicada a 66 estudiantes de la UTN, en donde se obtuvo una calificación satisfactoria del 82,39%. Con una eficiencia del 61%, se evidencia que hay margen de mejora en la experiencia de usuario para que sea más intuitiva para usuarios sin experiencia.

II DESARROLLO

El presente capítulo describe el proceso de desarrollo de la Fase 2 del portafolio docente de la aplicación móvil institucional UTN-Móvil, siguiendo la metodología ágil Scrum. En la **Figura 7** se presentan las tres fases propuestas en la guía metodológica de Huertas [5], las cuales se tomaron como referencia para la implementación del marco de trabajo Scrum en este trabajo de titulación.

Figura 7 Diagrama de fases de Scrum



Fuente: Propia

El desarrollo se estructuró en cinco sprints, incluyendo un sprint inicial de preparación (Sprint 0), cada uno con objetivos específicos alineados con las historias de usuario definidas en conjunto con el equipo Scrum. A lo largo de este capítulo se detalla la planificación, ejecución y resultados obtenidos en cada iteración, evidenciando el cumplimiento de los criterios de aceptación establecidos y la evolución incremental del producto.

2.1 Etapa de pre-juego

2.1.1 Definición del equipo Scrum

En esta sección se define los roles, los integrantes y sus responsabilidades en el desarrollo del proyecto de software. En la **Tabla 2** se puede observar la definición del equipo Scrum.

Tabla 2 Integrantes del Equipo Scrum

Rol	Nombre
Product Owner	Ph.D. José Antonio Quiña Mera
Scrum Master	Est. Diego David Recalde Varela
Equipo de desarrollo	
Stakeholder	Dirección de Desarrollo Tecnológico e Informático (DDTI)

2.1.2 Levantamiento de requisitos

En base a la planificación realizada en la Fase 1 del portafolio docente en el proyecto de investigación UTN-Móvil [3], se definieron nuevos requerimientos con el equipo Scrum y stakeholders de acuerdo a los lineamientos institucionales de la UTN. Los cuales establecieron los requisitos necesarios para la Fase 2 del portafolio docente, los mismos consisten en la implementación de nuevas funcionalidades para expandir y enriquecer el servicio. Conforme al marco de trabajo Scrum, se definieron las diferentes historias de usuario que cubren las nuevas funcionalidades y se alinean con el alcance definido previamente.

En la **Tabla 3** se puede observar la planificación realizada de todos los sprints que cubren el desarrollo de este trabajo de titulación, siguiendo el enfoque ágil de Scrum.

Tabla 3 Planificación General de Sprints

Sprints	Actividad	Duración
Sprint 0	Levantar los requisitos de software. Clonar y configurar los proyectos de backend y frontón. Definir las historias de usuario.	1 semana (8h)
Sprint 1	Desarrollo de proyectos de investigación, pagos y creación de aula virtual.	4 semanas (78h)
Sprint 2	Desarrollo de producción científica, foro online y control de bienes	4 semanas (78h)
Sprint 3	Desarrollo de banco de preguntas	4 semanas (79h)
Sprint 4	Desarrollo de evaluación online	4 semanas (78h)

2.1.2.1 Historias de Usuario

Tabla 4 Historia de Usuario 1

Código: HU1	Título: Lista de convocatorias	Prioridad: Media
Descripción:	Como docente de la UTN, quiero filtrar y visualizar todas las convocatorias de investigación.	
Criterios de aceptación:	CA1: La lista de convocatorias debe mostrar el tema, estado, descripción, fecha inicial y final, responsable y lugar. CA2: Se debe permitir el filtrado de convocatorias por estado activo e inactivo, además de ordenarlo ascendente y descendentemente. CA3: Al seleccionar una convocatoria, se debe mostrar una vista individual con la misma desde la cual, adicionalmente, se podrán visualizar los proyectos de investigación de dicha convocatoria.	

Tabla 5 Historia de Usuario 2

Código: HU2	Título: Lista de proyectos de investigación	Prioridad: Media
Descripción:	Como docente de la UTN, quiero visualizar y filtrar el listado de todos mis proyectos de investigación de una convocatoria específica.	
Criterios de aceptación:	CA1: Tras haber seleccionado la convocatoria, puedo visualizar todos mis proyectos, los cuales deben mostrar su título, duración en meses, fecha de registro, inicio y fin, también su estado y observación si es que la tiene. CA2: Se debe permitir el filtrado de proyectos de investigación por fecha mínima y máxima, además de ordenarlo ascendente y descendientemente. CA3: Al seleccionar un proyecto de investigación, se abrirá una nueva vista con los detalles de este.	

Tabla 6 Historia de Usuario 3

Código: HU3	Título: Detalle de proyecto de investigación	Prioridad: Media
Descripción:	Como docente de la UTN, quiero observar los detalles y secciones de un proyecto de investigación.	
Criterios de aceptación:	CA1: Se deben mostrar los diferentes los datos generales del proyecto de investigación, como lo son: título, convocatoria, tipo, países, ciudades, facultad, carrera, fecha de registro, inicio y fin, duración en meses, campo amplio, específico y detallado, grupo y red de investigación, descripción de articulación con docencia y vinculación, problema, metodología, justificación, objetivo general, bibliografía y observación. CA2: Se debe mostrar las diferentes secciones que tiene un proyecto de investigación, las cuales son: objetivos específicos, instituciones asociadas, estrategias, equipo de trabajo, estrategias de producción científica, infraestructura.	

Tabla 7 Historia de Usuario 4

Código: HU4	Título: Pagos	Prioridad: Media
Descripción:	Como docente de la UTN, quiero observar y filtrar el historial de todos mis pagos y transacciones hechas en la UTN.	
Criterios de aceptación:	CA1: El listado de pagos debe mostrar datos generales como: fecha de emisión, fecha de canjeo, estado, número de transacción, total de la transacción. CA2: Cada pago debe mostrar el listado de conceptos con su respectiva cantidad, descripción y subtotal. CA3: Cada pago debe tener un botón para acceder a la factura de la transacción en PDF. CA4: Se puede filtrar el listado de pagos por estado y total, además de ordenarlo ascendente y descendentemente.	

Tabla 8 Historia de Usuario 5

Código: HU5	Título: Creación de aula virtual	Prioridad: Alta
Descripción:	Como docente de la UTN, al inicio de un nuevo ciclo académico (semestre), quiero crear el aula virtual de las diferentes materias que se me han asignado en el nuevo ciclo.	
Criterios de aceptación:	CA1: Al acceder al registro académico, una vez seleccionado el nuevo ciclo académico, quiero crear el aula virtual de una materia sin necesidad de hacerlo primeramente en el SIIU. CA2: Al seleccionar la opción de crear el aula virtual, debe mostrarse un mensaje de confirmación antes de realizar la creación. CA3: Una vez creada el aula virtual, puedo acceder a la misma y visualizar sus diferentes apartados tales como actividades, recursos, notas y banco de preguntas.	

Tabla 9 Historia de Usuario 6

Código: HU6	Título: Lista de obras de producción científica	Prioridad: Media
Descripción:	Como docente de la UTN, quiero visualizar el listado de mis obras registradas en la sección de producción científica.	
Criterios de aceptación:	CA1: Al acceder a la sección de producción científica, puedo escoger entre todos los tipos de obras para acceder a su respectivo listado: artículos, libros, capítulos de libro, ponencias, y obras artísticas. CA2: Cada obra en el listado debe mostrar datos básicos que comparten todos los tipos de obra como: título, tipo de obra, fecha de registro, fecha de publicación, estado, si tiene filiación con la UTN y su estado de validación. CA3: Según el tipo de obra, cada una debe mostrar campos únicos correspondientes a su categoría, tales como base de datos indexada, idioma, ciudad, país, tipo de obra artística, etc. CA4: Al seleccionar una obra, accederé a una página con los detalles de esta.	

Tabla 10 Historia de Usuario 7

Código: HU7	Título: Detalle de obra de producción científica	Prioridad: Media
Descripción:	Como docente de la UTN, quiero observar los detalles y secciones de una obra de producción científica.	
Criterios de aceptación:	CA1: Tras seleccionar una obra desde el listado previo, se abrirá una página con los datos generales, autores, y documentos de la obra. CA2: En la sección de datos generales se mostrarán los siguientes campos, comunes a todos los tipos de obra: título, tipo de obra, fecha de registro, fecha de publicación, estado, filiación con la UTN, estado de validación, área de investigación, línea de investigación, campo amplio, campo específico, campo detallado, proyecto de investigación asociado y observaciones.' CA3: En la sección de datos generales, según el tipo de obra, cada una debe mostrar campos únicos correspondientes a su categoría, tales como: base de datos indexada, editorial, link, keywords, número de páginas, idioma, ciudad, país, tipo de obra artística, etc. CA4: En la sección de autores, se mostrará un listado con los siguientes datos: nombres, cédula, si es autor externo o principal. CA5: En la sección de documentos, según el tipo de obra, cada una debe mostrar documentos específicos correspondientes a su categoría disponibles para descargar, tales como: documento completo, documento de publicación, informes 1 y 2 de pares, certificado de ponencia, documento de SENADI, etc.	

Tabla 11 Historia de Usuario 8

Código: HU8 Título: Foro general		Prioridad: Baja
Descripción:	Como docente de la UTN, quiero visualizar y participar en el foro general de una materia.	
Criterios de aceptación:	CA1: Tras acceder al aula virtual de un curso, debe estar disponible el apartado de foro general. CA2: Si existen respuestas, deben mostrarse las diferentes participaciones tanto de docentes como de estudiantes, incluyendo datos como descripción, fecha de creación y autor. CA3: Puedo agregar nuevas respuestas; el texto de la descripción debe permitir formato enriquecido. CA4: Puedo editar mis respuestas. CA5: Puedo eliminar mis respuestas; al seleccionar la opción de eliminar, debe mostrarse un mensaje de confirmación antes de realizar la eliminación.	

Tabla 12 Historia de Usuario 9

Código: HU9	Título: Foro de actividad	Prioridad: Media
Descripción:	Como docente de la UTN, quiero visualizar y participar en el foro de una actividad académica.	
Criterios de aceptación:	CA1: Una vez dentro del aula virtual de un curso, al seleccionar una actividad, si es de tipo foro dentro de la plataforma institucional, debe estar disponible la opción para acceder al foro. CA2: Si existen respuestas, deben mostrarse las diferentes participaciones tanto de docentes como de estudiantes, incluyendo datos como descripción, fecha de creación y autor. CA3: Puedo agregar nuevas respuestas; el texto de la descripción debe permitir formato enriquecido. CA4: Puedo editar mis respuestas. CA5: Puedo eliminar mis respuestas; al seleccionar la opción de eliminar, debe mostrarse un mensaje de confirmación antes de realizar la eliminación.	

Tabla 13 Historia de Usuario 10

Código: HU10	Título: Control de bienes	Prioridad: Media
Descripción:	Como docente de la UTN, quiero visualizar y filtrar el listado de bienes y activos que tengo asignados.	
Criterios de aceptación:	CA1: La lista de activos debe incluir los siguientes datos: código, nombre, observación, precio de compra, fecha de compra, fotografía y condición actual. CA2: Dentro de la sección de control de bienes, la app debe permitir la descarga del formato de Word "Modelo de Actas". CA3: Se puede generar y visualizar el reporte de control de bienes, según el tipo escogido: activos o bienes. CA4: Se puede filtrar el listado de activos por condición, rango de precio de compra (mínimo y máximo), así como ordenarlo ascendente y descendentemente por la fecha de compra.	

Tabla 14 Historia de Usuario 11

Código: HU11	Título: Lista de preguntas	Prioridad: Alta
Descripción:	Como docente de la UTN, quiero visualizar y filtrar el listado de preguntas del aula virtual de una materia.	
Criterios de aceptación:	CA1: Las preguntas deben incluir datos como: enunciado, tipo de pregunta, nivel de dificultad, imagen y fecha de creación. CA2: Cada pregunta debe proporcionar las siguientes opciones: ver(detalle), editar y eliminar. CA3: Desde la página de listado de preguntas, deben estar disponible el acceso a la funcionalidad de crear e importar preguntas. CA4: Se puede filtrar el listado de preguntas por contenido, tipo, rango de dificultad.	

Tabla 15 Historia de Usuario 12

Código: HU12	Título: Detalle de pregunta	Prioridad: Media
Descripción:	Como docente de la UTN, quiero observar los detalles de una pregunta, incluyendo sus alternativas de respuesta.	
Criterios de aceptación:	CA1: Se deben incluir los siguientes datos de la pregunta: enunciado, tipo de pregunta, nivel de dificultad, imagen y fecha de creación. CA2: Se deben mostrar las alternativas de respuesta de la pregunta cuando esta no sea de tipo ensayo. Para preguntas de respuesta única u opción múltiple, se deben mostrar las alternativas con sus respectivos enunciados, indicando claramente la(s) respuesta(s) correcta(s). Para preguntas de emparejamiento de columnas, se deben mostrar los enunciados de ambas columnas junto con sus correspondencias. CA3: Desde la página de detalle de la pregunta, debe ser posible editarla o eliminarla.	

Tabla 16 Historia de Usuario 13

Código: HU13	Título: Crear pregunta	Prioridad: Alta
Descripción:	Como docente de la UTN, quiero crear una nueva pregunta mediante un formulario.	
Criterios de aceptación:	CA1: El formulario para crear una pregunta debe contar con los siguientes campos obligatorios: enunciado (con editor enriquecido), contenido, tipo, y nivel de dificultad. El formulario debe permitir ingresar una imagen de forma opcional. CA2: No se puede agregar una pregunta si el ciclo académico de la materia es inactivo. CA3: Según el tipo de pregunta seleccionado, el formulario debe mostrar dinámicamente los campos. Si es respuesta única u opción múltiple: ingreso de alternativas y selección de la(s) correcta(s). Si es emparejamiento de columnas, ingreso de enunciados para ambas columnas y su correspondencia. Y si es ensayo, no se deben mostrar campos de alternativas. CA4: La app debe validar que todos los campos obligatorios estén completos antes de permitir el guardado. Si la pregunta a crear no es tipo ensayo, se deben realizar validaciones antes de guardar la pregunta, como verificar que en respuesta única u opción múltiple al menos exista una respuesta correcta. CA5: Al guardar la pregunta correctamente, la app debe registrar la fecha de creación y redirigir a la página de listado de preguntas.	

Tabla 17 Historia de Usuario 14

Código: HU14	Título: Editar pregunta	Prioridad: Media
Descripción:	Como docente de la UTN, quiero editar una pregunta existente mediante un formulario.	
Criterios de aceptación:	CA1: Desde la página de detalle de la pregunta y en el listado de preguntas, debe existir una opción para editar una pregunta específica. CA2: No se puede editar una pregunta si el ciclo académico de la materia es inactivo. CA3: Se debe mostrar un formulario precargado con la información actual de la pregunta y alternativas de respuesta. CA4: El formulario debe permitir modificar los mismos campos definidos para la creación de la pregunta y alternativas de respuesta. Las validaciones deben ser las mismas que en la creación de la pregunta. CA5: Al guardar los cambios, la información de la pregunta debe actualizarse y reflejarse en la página de listado de preguntas.	

Tabla 18 Historia de Usuario 15

Código: HU15	Título: Eliminar pregunta	Prioridad: Media
Descripción:	Como docente de la UTN, quiero eliminar una pregunta perteneciente al aula virtual de una materia.	
Criterios de aceptación:	CA1: Desde la página de listado de preguntas y en la de detalle, debe existir una opción para eliminar. CA2: No se puede eliminar una pregunta si el ciclo académico de la materia es inactivo o si está ya pertenece al cuestionario de una evaluación online. CA3: Al seleccionar la opción de eliminar, la app debe solicitar una confirmación explícita al docente. CA4: Si se confirma la eliminación, la pregunta debe marcarse como eliminada (eliminación lógica) y no debe ser visible en el banco de preguntas.	

Tabla 19 Historia de Usuario 16

Código: HU16	Título: Importar preguntas	Prioridad: Media
Descripción:	Como docente de la UTN, quiero importar preguntas desde otra materia o contenido para reutilizarlas en mi aula virtual sin recrearlas.	
Criterios de aceptación:	CA1: Al seleccionar “Importar” desde la página de lista de preguntas, se visualiza el flujo de importación por pasos con selectores de ciclo, materia y contenido. CA2: No se puede importar preguntas si el ciclo académico de la materia es inactivo. CA3: En el paso de origen, el docente elige un contenido en base al ciclo y materia seleccionados. En el paso de destino, el docente elige el contenido de su materia actual en donde se importarán las preguntas. CA4: En el paso final, el docente elige el modo: “individual” (puede ver preguntas, seleccionar/deseleccionar, seleccionar todas las no importadas) o “todo” (ve un resumen de cuántas se importarán). CA5: La acción de importar solo se habilita en el último paso cuando las validaciones se cumplen (en “individual”, hay preguntas seleccionadas). Al finalizar la importación, la app redirige al listado de preguntas en donde se reflejan las preguntas recién importadas.	

Tabla 20 Historia de Usuario 17

Código: HU17	Título: Agregar preguntas a evaluación	Prioridad: Alta
Descripción:	Como docente de la UTN que administra una evaluación en línea, quiero agregar preguntas existentes a una evaluación en línea.	
Criterios de aceptación:	CA1: Al seleccionar una actividad desde el listado de actividades del aula virtual o desde el calendario general, si la actividad es de tipo evaluación online y la evaluación es de tipo manual, la app debe mostrar la opción "Agregar Preguntas". CA2: Al acceder a la opción "Agregar Preguntas", la app debe mostrar una página dividida en dos secciones. CA3: En "preguntas agregadas" debe visualizarse un listado de las preguntas que ya forman parte del cuestionario, desde el cual se pueden seleccionar una o más preguntas para eliminarlas de la evaluación. CA4: En "preguntas disponibles" debe visualizarse un listado de preguntas disponibles en el banco de preguntas, desde el cual se pueden seleccionar una o más preguntas para agregarlas a la evaluación. CA5: Deben estar disponibles las acciones de eliminar todas las preguntas, así como restablecer el tiempo de evaluación de los estudiantes.	

Tabla 21 Historia de Usuario 18

Código: HU18	Título: Sortear preguntas de evaluación	Prioridad: Alta
Descripción:	Como docente de la UTN que administra una evaluación en línea, quiero sortear preguntas existentes a una evaluación en línea.	
Criterios de aceptación:	CA1: Al seleccionar una actividad desde el listado de actividades del aula virtual o desde el calendario general, si la actividad es de tipo evaluación online y la evaluación es de tipo automática, la app debe mostrar la opción "Sortear Preguntas". CA2: Al acceder a la página de sorteo, cuando aún no se haya realizado el mismo, se debe mostrar un formulario de configuración que permita definir el contenido, la dificultad mínima y máxima, la opción de generar un cuestionario diferente para cada estudiante y el número de preguntas; en función de estos parámetros, el sistema debe mostrar el total de preguntas disponibles en el banco de preguntas. CA3: La acción de sortear solo se permite si se han configurado todos los parámetros correctamente y se ha escrito un número de preguntas válido. CA4: Finalizado el sorteo, se muestra la vista de estudiantes con el total de preguntas asignadas por estudiante; cada estudiante despliega sus preguntas sorteadas con enunciado, tipo y nivel de dificultad. CA5: Deben estar disponibles las acciones de eliminar todas las preguntas, así como restablecer el tiempo de evaluación de los estudiantes.	

Tabla 22 Historia de Usuario 19

Código: HU19	Título: Revisión de evaluación online	Prioridad: Alta
Descripción:	Como docente de la UTN, quiero revisar el intento de evaluación online de un estudiante para analizar sus respuestas y nota obtenida.	
Criterios de aceptación:	CA1: Al acceder al apartado de calificaciones de una actividad de tipo evaluación online, se debe permitir revisar el intento de evaluación de todos los estudiantes. CA2: Deben visualizarse todas las preguntas de la evaluación con: enunciado, nota, tipo, nivel de dificultad y alternativas de respuesta si es no es de tipo ensayo. CA3: Para preguntas de tipo ensayo, debe indicarse que requieren calificación manual. CA4: Debe mostrarse el puntaje total obtenido por el estudiante en la evaluación.	

Tabla 23 Historia de Usuario 20

Código: HU20	Título: Calificación de preguntas de ensayo	Prioridad: Alta
Descripción:	Como docente de la UTN, quiero calificar manualmente las preguntas de tipo ensayo para completar la nota final de la evaluación del estudiante.	
Criterios de aceptación:	CA1: Se debe permitir al docente calificar únicamente las preguntas de tipo ensayo del intento de evaluación de un estudiante. CA2: Cada pregunta de tipo ensayo solo debe poder calificarse con los valores correcta (1) o incorrecta (0). CA3: No se puede calificar preguntas si el ciclo académico de la materia es inactivo. CA4: Al registrar la calificación de una pregunta de tipo ensayo, la app debe recalcular automáticamente la nota total de la evaluación.	

2.1.3 Product backlog

Uno de los pilares clave de la metodología ágil Scrum es el Product Backlog, el cual consiste en una lista dinámica y ordenada de ítems (historias de usuario) necesarios para el desarrollo de un proyecto. El mismo garantiza la transparencia y el enfoque estratégico al desglosar objetivos complejos en tareas manejables mediante un proceso de refinamiento y mejora continua. Asimismo, el Product Owner guía la priorización de valores y compensaciones, permitiendo que el producto crezca de manera adaptativa y alineada con las necesidades reales del negocio [58].

Una vez definidos todos los requisitos, se ordenan y estiman en el Product Backlog. Existen diferentes técnicas de estimación, una de ellas son los puntos de historia (Story Points), los cuales son métricas de complejidad utilizadas para estimar el esfuerzo, costo y duración de cada una de las tareas del Product Backlog. Generalmente los valores de los puntos de historia suelen realizarse siguiendo la serie de Fibonacci [71]. Los resultados de la estimación pueden observarse en la **Tabla 24**.

Tabla 24 Product Backlog

Índice	Código	Descripción	Puntos (SP)
1	HU1	Lista de convocatorias	3
2	HU2	Lista de proyectos de investigación	3
3	HU3	Detalle de proyecto de investigación	5
4	HU4	Pagos	5
5	HU5	Creación de aula virtual	5
6	HU6	Lista de obras de producción científica	5
7	HU7	Detalle de obra de producción científica	5
8	HU8	Foro general	8
9	HU9	Foro de actividad	3
10	HU10	Control de bienes	5
11	HU11	Lista de preguntas	3
12	HU12	Detalle de pregunta	2
13	HU13	Crear pregunta	13
14	HU14	Editar pregunta	8
15	HU15	Eliminar pregunta	2

16	HU16	Importar preguntas	13
17	HU17	Agregar preguntas a evaluación	8
18	HU18	Sortear preguntas de evaluación	8
19	HU19	Revisión de evaluación online	5
20	HU20	Calificación de preguntas de ensayo	3

2.2 Etapa de juego

2.2.1 Sprint 0

El sprint 0 puede considerarse como la fase de pre-desarrollo, en la cual arranca el proyecto Scrum. En la misma se realiza la configuración del entorno de desarrollo, creación del Product Backlog, y más importante la definición de la arquitectura de software. En la arquitectura se establecen todos los componentes de software que participan e interactúan entre sí para proveer el servicio del portafolio docente en la aplicación móvil.

2.2.1.1 Definición de arquitectura de microservicios

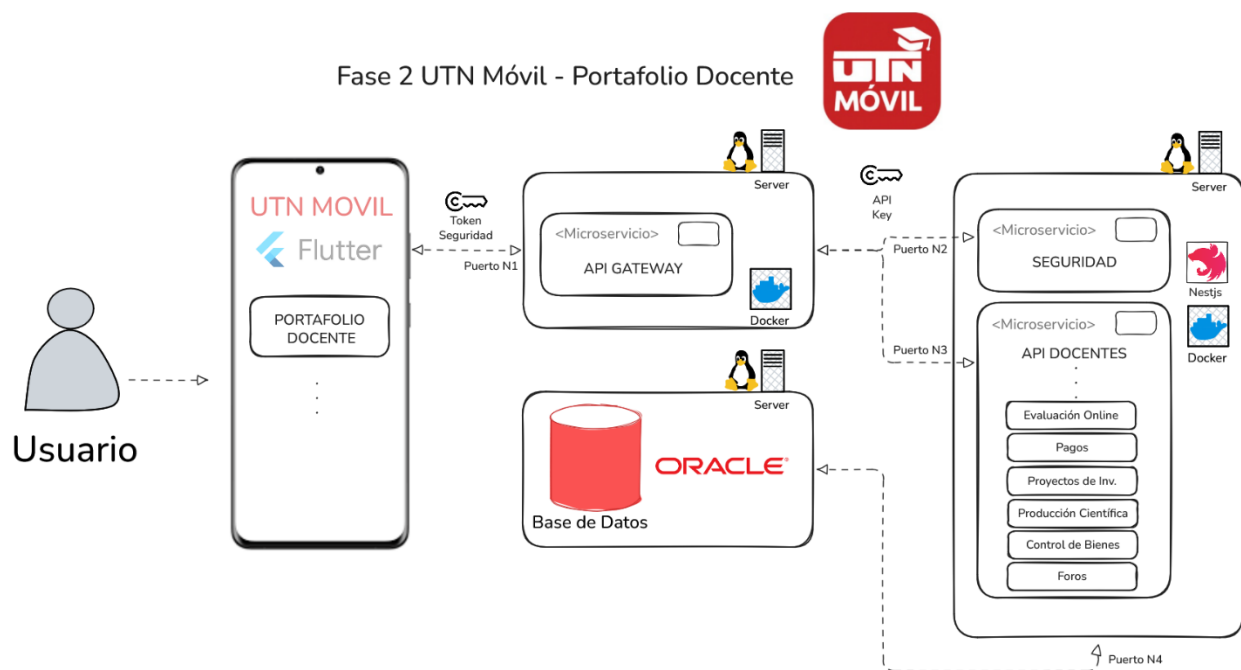
Debido a que el presente trabajo de titulación consiste en la continuación de la Fase 1 implementada por el Ing. Franklin Enriquez, se ha mantenido la misma arquitectura de microservicios la cual ha demostrado ser segura y eficiente. La arquitectura de UTN-Móvil está basada en diferentes microservicios contenerizados que se comunican entre sí para recibir y enviar información al cliente.

El backend fue desarrollado a través del framework NestJS, en el cual se utilizó el lenguaje de programación TypeScript. Este backend proporciona una API Rest con múltiples endpoints que consultan e interactúan con el cliente Flutter y con la base de datos institucional Oracle, mediante la librería TypeORM. De tal manera que se garantiza una fuerte integración con el portafolio docente del SIIU, la aseguración de la consistencia de datos permite la coexistencia entre el SIIU y UTN-Móvil.

Como se observa en la **Figura 8**, la arquitectura de UTN-Móvil utiliza un API Gateway para centralizar el control de acceso y la validación de identidad del docente. Este microservicio recibe las solicitudes del cliente Flutter, autentica al usuario y redirige la operación al microservicio correspondiente. El ciclo concluye con la entrega de datos en formato JSON, un

método que optimiza la comunicación entre el backend y la interfaz móvil bajo estándares de alta seguridad [3]. Actualmente, en el mismo servidor Linux se ejecutan los diferentes microservicios implementados en la Fase 1, tales como: portafolio estudiante, portafolio docente, seguridad, notificaciones y Chat UTN.

Figura 8 Arquitectura de microservicios UTN-Móvil



Fuente: Propia

2.2.1.2 Entorno de desarrollo

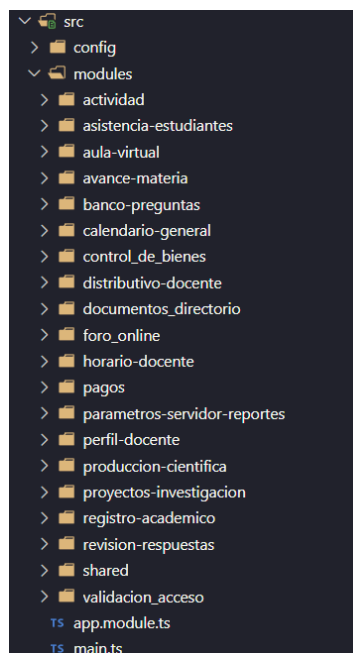
Al trabajar en base al servicio del portafolio docente del SIIU que fue desarrollado en Oracle APEX, fue crucial para este trabajo de titulación el acceso al entorno de desarrollo. Desde el mismo se pudo ver en detalle cómo están estructuradas y codificadas las diferentes funcionalidades del portafolio docente que se van a implementar en la presente Fase 2. De esta manera, fue posible estudiar las consultas SQL y PL/SQL utilizadas por los distintos componentes, así como las reglas de negocio asociadas a cada funcionalidad. Con esta información valiosa, se implementaron las nuevas funcionalidades de la Fase 2 en la aplicación móvil, manteniendo fidelidad a las funcionalidades originales desarrolladas en APEX, respetando la base de datos y, a su vez, mejorando su funcionamiento.

El acceso al entorno de desarrollo fue proporcionado por los ingenieros del DDTI, quienes facilitaron una conexión VPN segura que permitió trabajar de forma remota en el proyecto, así como el acceso a las credenciales de la base de datos y de Oracle APEX.

2.2.1.3 Estructura del proyecto backend

La estructura del proyecto backend en NestJS está basada en módulos. Como se muestra en la **Figura 9**, el directorio `src` contiene los archivos de autenticación y configuración, así como todos los módulos del portafolio docente, el módulo raíz `app.module.ts` y el punto de entrada de la aplicación `main.ts`.

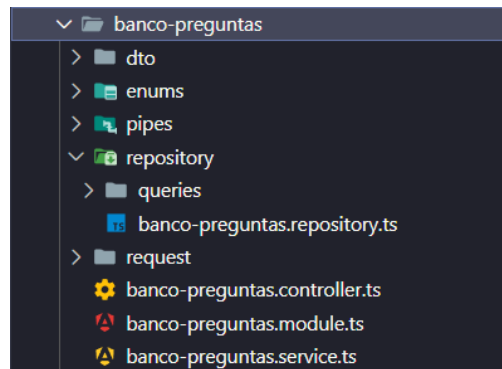
Figura 9 Estructura del microservicio docente



Fuente: Propia

La estructura típica de un módulo del backend NestJS contiene las carpetas mostradas en la **Figura 10**. La arquitectura de cada módulo se conforma por estas tres capas: repository, service y controller.

Figura 10 Estructura de un módulo de NestJS



Fuente: Propia

Donde las carpetas del módulo representan las siguientes funcionalidades:

- **dto:** Contiene clases de TypeScript que representan objetos de transferencia de datos (Data Transfer Object). Estos se utilizan para definir y validar la estructura de los datos que se intercambian entre las distintas capas de la aplicación, así como para mapear los campos obtenidos desde la base de datos a través del *repository*.
- **pipes:** Contiene *pipes* personalizados que permiten realizar validaciones y transformaciones a los datos entrantes de las solicitudes antes de que sean procesados por los *controllers*.
- **enums:** Contiene archivos que definen estructuras de enumeración, las cuales representan un conjunto de valores constantes. Son útiles para modelar tipos, categorías, roles o estados, y contribuyen a mejorar la mantenibilidad y el control del código.
- **repository:** Contiene una carpeta de *queries* y el archivo del *repository*. Esta capa es la encargada de interactuar directamente con la base de datos y define los métodos que ejecutan las consultas SQL declaradas en la carpeta *queries*.

Mientras que los archivos principales de un módulo realizan las siguientes funcionalidades:

- **service:** Inyecta al *repository* y se encarga de aplicar la lógica de negocio de la aplicación. Asimismo, gestiona la captura de errores mediante bloques *try/catch*, permitiendo el envío de excepciones con mensajes personalizados.

- **controller:** Define los distintos *endpoints* del módulo que expone la API, aplica los *pipes* correspondientes, gestiona las solicitudes HTTP y envía las respuestas al frontend.
- **module:** Define y registra los *controllers* y *providers* que requiere el módulo. Además, exporta el módulo para que pueda ser utilizado por el módulo raíz en el archivo *app.module.ts*.

2.2.1.4 Estructura del proyecto frontend

El proyecto frontend Flutter **utn_movil_app** representa el punto central donde convergen todos los microservicios que conforman el proyecto institucional UTN-Móvil. A través de esta aplicación se ofrecen diversos servicios, como el portafolio estudiantil, el portafolio docente, el chat UTN y la gestión de eventos, entre otros.

La aplicación móvil multiplataforma se fundamenta en una arquitectura modular y escalable, siguiendo los principios de arquitectura limpia. El desarrollo de los diferentes servicios se centraliza en el directorio *lib*, donde se articulan tanto la lógica de negocio como los componentes de la interfaz de usuario.

Los archivos de componentes principales de ejecución son:

- **main.dart:** Constituye el punto de entrada principal del software. Su función primordial es la orquestación inicial del sistema, encargándose de la configuración de variables de entorno y la inicialización de los servicios necesarios para el arranque.
- **app.dart:** Define el *widget* raíz de la aplicación. En este archivo se centralizan las configuraciones globales de la experiencia de usuario, incluyendo el sistema de rutas, la gestión de temas visuales (estilos), la localización y el soporte multi-idioma.

Para garantizar un sistema robusto y mantenible, se han integrado las siguientes herramientas y librerías:

- **injection_container.dart:** Actúa como el motor de inyección de dependencias mediante la librería *GetIt*. Su diseño permite una gestión modular de los servicios, facilitando la integración de los diferentes microservicios de UTN-Móvil.
- **Dio:** Para la comunicación eficiente con las APIs del backend mediante solicitudes HTTP.

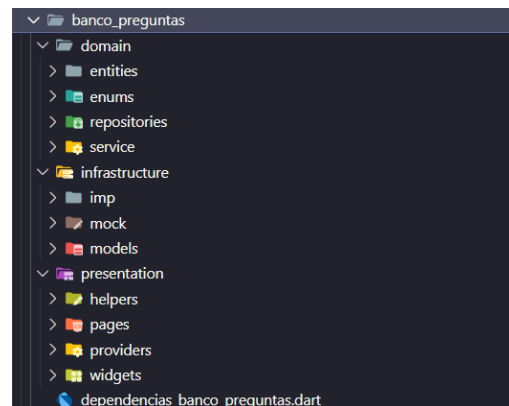
- **Provider:** Utilizado para la gestión de estados a lo largo de la aplicación.
- **GoRouter:** Encargado de administrar el flujo de navegación de forma declarativa.

Las siguientes carpetas contienen funcionalidades para la organización y reutilización de código:

- **shared:** Directorio diseñado para alojar funciones, utilitarios y *widgets* genéricos que son utilizados por múltiples módulos de la aplicación, facilitando la reutilización y mantenibilidad de código.
- **config:** Carpeta que contiene las configuraciones globales, donde se definen constantes, esquemas de navegación y los parámetros de diseño que rigen el comportamiento de la aplicación.

En la **Figura 11** se puede visualizar la estructura definida de un módulo del servicio del portafolio docente en el proyecto frontend.

Figura 11 Estructura del cliente Flutter



Fuente: Propia

El desarrollo se estructuró bajo el patrón de arquitectura limpia, segmentando las responsabilidades en tres capas fundamentales que garantizan la división de responsabilidades facilitan la mantenibilidad de la aplicación:

1. Capa de dominio

Considerada el núcleo de la aplicación, esta capa es independiente de agentes externos y define la lógica de negocio. Se compone de:

- **Entities:** Clases de Dart que consisten en estructuras de datos base que representan los objetos de negocio requeridos para las interacciones con el backend.
- **Queries:** Definen la estructura de los parámetros necesarios para las consultas de datos.
- **Repositories y Services:** Establecen las interfaces y contratos lógicos que rigen el comportamiento funcional de la aplicación.

2. Capa de infraestructura

Esta capa es la encargada de la comunicación con servicios externos y el backend. Su implementación se organiza en las siguientes subcarpetas:

- **imp (Implementations):** Contiene la ejecución concreta de los repositorios definidos en la capa de dominio.
- **Models:** Se especializa en el mapeo y la serialización de datos (JSON) hacia las entidades de dominio, asegurando la integridad de la información.
- **Mocks:** Proporciona datos simulados para las peticiones al backend, facilitando las pruebas unitarias y el desarrollo en entornos aislados sin depender del backend.

3. Capa de presentación

Gestiona la interacción directa con el usuario y la renderización de la interfaz. Su estructura se divide en:

- **Providers:** Actúan como gestores de estado, mediando entre la interfaz de usuario y la capa de dominio para procesar las peticiones y actualizar la vista de forma reactiva.
- **Widgets:** Componentes visuales personalizados y modulares que se combinan para construir la interfaz.
- **Pages:** Representan los puntos de entrada para cada requisito. Estas delegan la lógica de estado a los *providers*, enfocándose exclusivamente en la disposición visual de los *widgets* para conformar vistas completas y organizadas.

Como mecanismo de cierre para la integración de las capas, se utiliza el contenedor de servicios *GetIt*. Este se encarga de la inicialización y gestión centralizada de las instancias de los *providers*, *services* y *repositories*. Este enfoque garantiza una inyección de dependencias eficiente, permitiendo que el sistema sea escalable y facilitando el mantenimiento al controlar el ciclo de vida de los componentes desde un único punto de registro.

2.2.2 Sprint 1

2.2.2.1 Planificación del sprint 1

Conforme a la reunión realizada con el equipo Scrum, se asignaron las siguientes historias de usuario para el primer Sprint Backlog, como se puede ver en la **Tabla 25**.

Tabla 25 Sprint backlog 1

Código	Nombre	Tarea	Horas
HU1	Lista de convocatorias	Desarrollo de endpoint GET	1
		Desarrollo de pruebas unitarias	1
		Desarrollo de interfaz	2
HU2	Lista de proyectos de investigación	Desarrollo de endpoint GET	2
		Desarrollo de pruebas unitarias	1
		Desarrollo de interfaz	2
HU3	Detalle de proyecto de investigación	Desarrollo de endpoint GET	6
		Desarrollo de pruebas unitarias	2
		Desarrollo de interfaz	8
HU4	Pagos	Desarrollo de endpoint GET	4
		Desarrollo de pruebas unitarias	1
		Desarrollo de interfaz	3
HU5	Creación de aula virtual	Desarrollo de endpoint POST	4
		Desarrollo de pruebas unitarias	1
		Desarrollo de interfaz	1
TOTAL			40

2.2.2.2 Incremento del sprint 1

En este sprint se implementó el módulo de proyectos de investigación. Al acceder al apartado de proyectos, se puede acceder a un listado de proyectos del docente “Mis Proyectos” o acceder a la sección de convocatorias para visualizar los proyectos que tiene el docente en cada convocatoria.

Figura 12 Opciones de proyectos de investigación

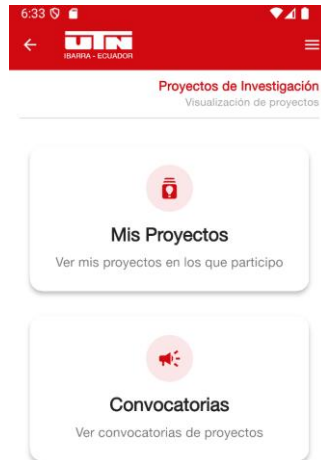


Figura 13 Mis proyectos de investigación



Figura 14 Listado de convocatorias



Figura 15 Listado de proyectos por convocatoria



Tanto los listados de proyectos como los de convocatorias cuentan con múltiples filtros, como rango de fechas u ordenamiento por más antiguos o más recientes.

Figura 16 Filtrado de proyectos

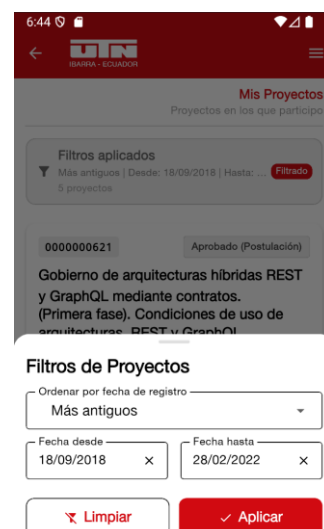
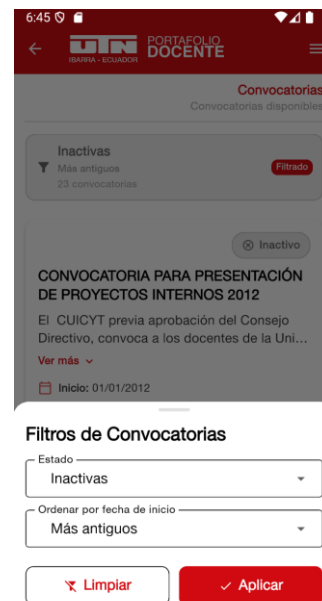


Figura 17 Filtrado de convocatorias



Al seleccionar la opción de visualizar un proyecto, se abrirá la página del detalle de este, la cual cuenta con múltiples pestañas (tabs) de datos relacionados al proyecto.

Figura 18 Datos generales de proyecto



Figura 19 Objetivos específicos del proyecto



Figura 20 Áreas de investigación y ejes integrados del proyecto

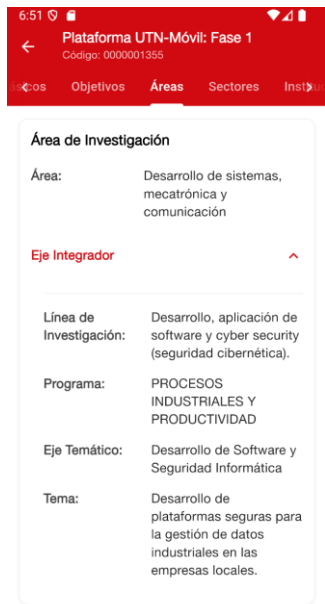


Figura 21 Sectores de impacto del proyecto



Figura 22 Instituciones asociadas al proyecto



Figura 23 Beneficiarios del proyecto



Figura 24 Equipo de trabajo del proyecto



Figura 25 Estrategias de socialización del proyecto

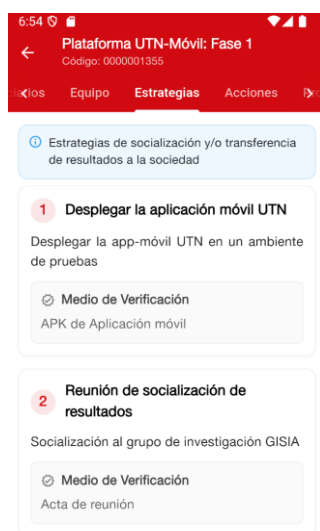


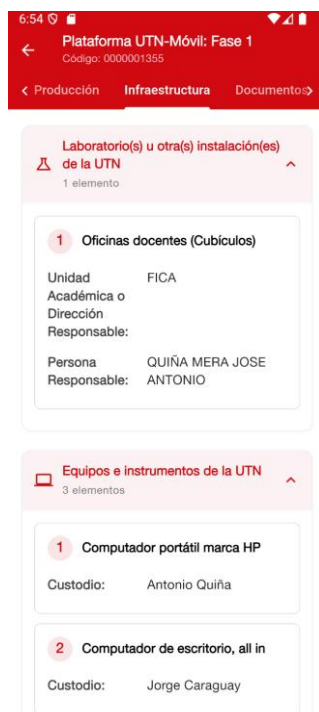
Figura 26 Acciones de divulgación del proyecto



Figura 27 Estrategias de producción científica del proyecto



Figura 28 Infraestructura del proyecto



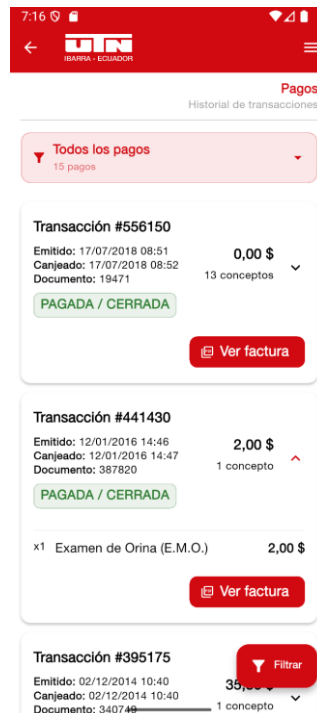
Como se observa en la **Figura 29**, desde la pestaña de documentos se pueden descargar al dispositivo móvil los diferentes documentos adjuntos al proyecto. Al descargar el documento se le preguntará a el docente la ruta en la que desea guardar el archivo.

Figura 29 Documentos del proyecto



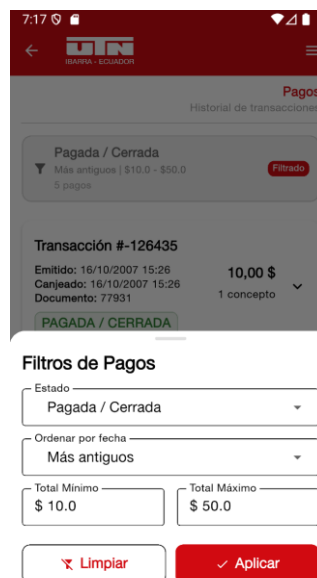
Finalizado el módulo de proyectos de investigación, se realizó el módulo de pagos, el cual consiste en la visualización del listado de transacciones con sus respectivos conceptos hechos por el docente. Además, se puede visualizar el reporte de cada pago.

Figura 30 Listado de pagos



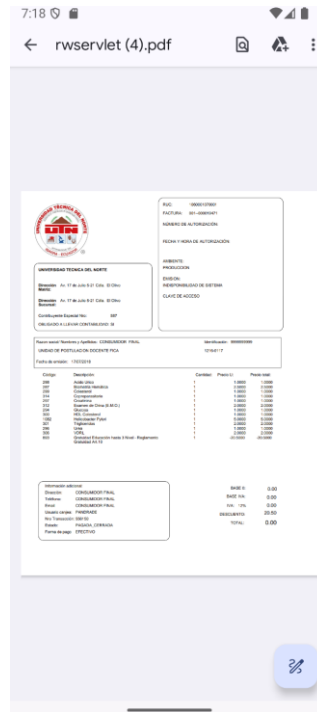
Así como en el listado de proyectos, también se pueden aplicar diferentes filtros al listado de pagos, tales como rango de total, estado y ordenamiento por más recientes y más antiguos.

Figura 31 Filtrado de pagos



Al seleccionar el botón de 'Ver Factura' se abrirá en el navegador el mismo tipo de reporte generado por el SIIU, el mismo se puede descargar como PDF.

Figura 32 Reporte de pago generado



Para finalizar este sprint, se implementó la funcionalidad de crear un aula virtual. La misma está disponible desde el registro académico y es accesible mediante el nuevo ícono de crear.

Figura 33 Nuevo ícono de crear aula virtual

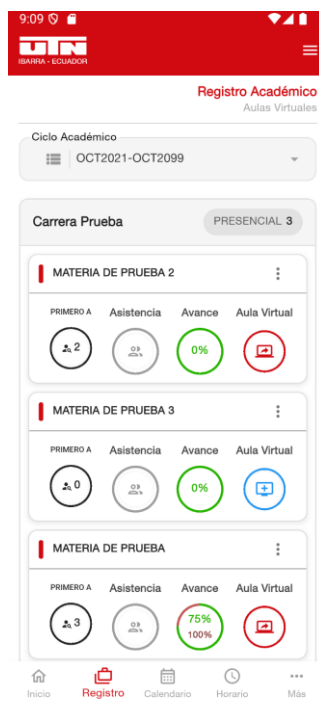


Figura 34 Confirmación de creación de aula virtual



2.2.2.3 Revisión del sprint 1

Teniendo en cuenta los criterios de aceptación de todas las historias de usuario cubiertas en este sprint, se realizó la revisión correspondiente. Como se observa en la **Tabla 26**, se detallan los elementos cumplidos.

Tabla 26 Revisión de éxito por criterios de aceptación del sprint 1

Código	Criterios de Aceptación	Cumple
HU1	CA1: La lista de convocatorias debe mostrar el tema, estado, descripción, fecha inicial y final, responsable y lugar.	SI
	CA2: Se debe permitir el filtrado de convocatorias por estado activo e inactivo, además de ordenarlo ascendente y descendientemente.	SI
	CA3: Al seleccionar una convocatoria, se debe mostrar una vista individual con la misma desde la cual, adicionalmente, se podrán visualizar los proyectos de investigación de dicha convocatoria.	SI
HU2	CA1: Tras haber seleccionado la convocatoria, puedo visualizar todos mis proyectos, los cuales deben mostrar su título, duración en meses, fecha de registro, inicio y fin, también su estado y observación si es que la tiene.	SI
	CA2: Se debe permitir el filtrado de proyectos de investigación por fecha mínima y máxima, además de ordenarlo ascendente y descendientemente.	SI
	CA3: Al seleccionar un proyecto de investigación, se abrirá una nueva vista con los detalles de este.	SI
HU3	CA1: Se deben mostrar los diferentes los datos generales del proyecto de investigación, como lo son: título, convocatoria, tipo, países, ciudades, facultad, carrera, fecha de registro, inicio y fin, duración en meses, campo amplio, específico y detallado, grupo y red de investigación, descripción de articulación con docencia y vinculación, problema, metodología, justificación, objetivo general, bibliografía y observación.	SI
	CA2: Se debe mostrar las diferentes secciones que tiene un proyecto de investigación, las cuales son: objetivos específicos, instituciones	SI

	asociadas, estrategias, equipo de trabajo, estrategias de producción científica, infraestructura	
HU4	CA1: El listado de pagos debe mostrar datos generales como: fecha de emisión, fecha de canjeo, estado, número de transacción, total de la transacción.	SI
	CA2: Cada pago debe mostrar el listado de conceptos con su respectiva cantidad, descripción y subtotal.	SI
	CA3: Cada pago debe tener un botón para acceder a la factura de la transacción en PDF.	SI
	CA4: Se puede filtrar el listado de pagos por estado y total, además de ordenarlo ascendente y descendientemente.	SI
HU5	CA1: Al acceder al registro académico, una vez seleccionado el nuevo ciclo académico, quiero crear el aula virtual de una materia sin necesidad de hacerlo primeramente en el SIIU.	SI
	CA2: Al seleccionar la opción de crear el aula virtual, debe mostrarse un mensaje de confirmación antes de realizar la creación.	SI
	CA3: Una vez creada el aula virtual, puedo acceder a la misma y visualizar sus diferentes apartados tales como actividades, recursos, notas y banco de preguntas.	SI

2.2.2.4 Retrospectiva del sprint 1

El sprint se completó exitosamente cumpliendo todas las historias de usuario planificadas. La implementación del módulo de proyectos de investigación representó un desafío importante por la complejidad de las consultas SQL y la cantidad de datos relacionados a visualizar. La experiencia adquirida al estudiar el código de Oracle APEX facilitó la comprensión de las reglas de negocio y permitió una implementación más fiel al SIIU. Se identificó la necesidad de mejorar la estimación de tiempos para módulos con múltiples pestañas y secciones de datos, ya que estos requieren más esfuerzo de desarrollo del inicialmente previsto.

2.2.3 Sprint 2

2.2.3.1 Planificación del sprint 2

Tabla 27 Sprint backlog 2

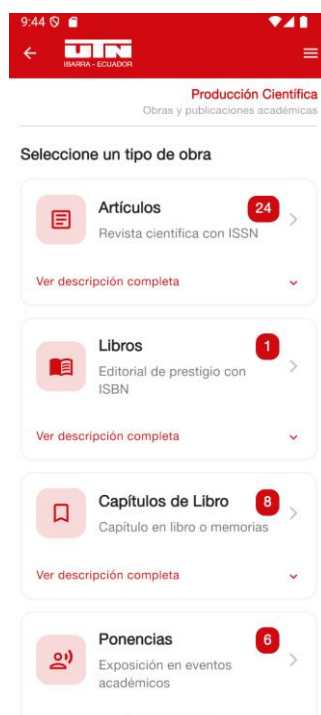
Código	Nombre	Tarea	Horas
HU6	Lista de obras de producción científica	Desarrollo de endpoint GET de artículos	2
		Desarrollo de endpoint GET de libros	1
		Desarrollo de endpoint GET de capítulos de libros	0.5
		Desarrollo de endpoint GET de ponencias	1
		Desarrollo de endpoint GET de obras artísticas	1
		Desarrollo de pruebas unitarias	2
		Desarrollo de interfaz	3
HU7	Detalle de obra de producción científica	Desarrollo de endpoint GET de detalle de artículo	2
		Desarrollo de endpoint GET de detalle de libro	1
		Desarrollo de endpoint GET de detalle de capítulos de libro	0.5
		Desarrollo de endpoint GET de detalle de ponencia	1
		Desarrollo de endpoint GET de detalle de obra artística	1
		Desarrollo de pruebas unitarias	2
		Desarrollo de interfaz	3
HU8	Foro general	Desarrollo de endpoint GET	2
		Desarrollo de endpoint DELETE	0.5
		Desarrollo de endpoint POST	3
		Desarrollo de endpoint UPDATE	3

		Desarrollo de pruebas unitarias	4
		Desarrollo de interfaz	10
HU9	Foro de actividad	Desarrollo de endpoint GET	1
		Desarrollo de endpoint DELETE	0.5
		Desarrollo de endpoint POST	1
		Desarrollo de endpoint UPDATE	1
		Desarrollo de pruebas unitarias	2
		Desarrollo de interfaz	3
HU10	Control de bienes	Desarrollo de endpoint GET	1
		Desarrollo de pruebas unitarias	1
		Desarrollo de interfaz	3
TOTAL			57

2.2.3.2 Incremento del sprint 2

El primer módulo implementado en el sprint 2 fue el de producción científica. Al acceder a este nuevo apartado, se puede escoger de entre diferentes tipos de obra; además, se indica el número de registros que tiene el docente por cada tipo de obra.

Figura 35 Tipos de obra de producción científica



Según el tipo de obra escogido, se mostrarán u ocultarán ciertos parámetros en cada obra del listado.

Figura 36 Lista de artículos



Figura 37 Lista de libros



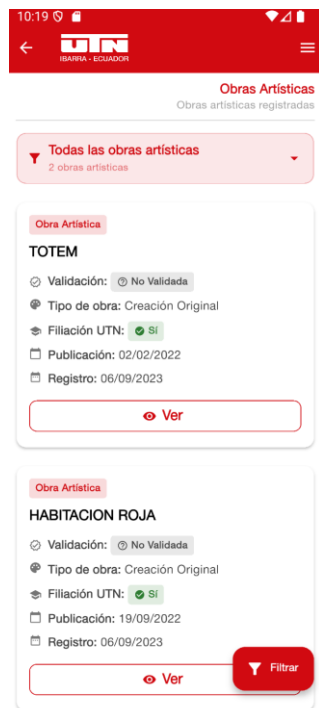
Figura 38 Lista de capítulos de libros



Figura 39 Lista de ponencias



Figura 40 Lista de obras artísticas



Todos los listados de los diferentes tipos de obra tienen opciones de filtro disponible, tales como rango de fecha de registro y ordenamiento por más recientes y antiguos. En la **Figura 41** se observa los filtros del listado de artículos.

Figura 41 Filtrado de artículos

Al seleccionar una obra, se abrirá una nueva página de detalle en la que se muestran diferentes pestañas de información, así como en la página de detalle de proyecto de investigación. En la pestaña de datos generales, se muestra y ocultan ciertos campos según el tipo de obra.

Figura 42 Datos generales de obra

10:17 Integración de componentes Enterprise Java Beans y Front End Angular para ... Código: 7697

< Datos Autores Documento >

Información General

Título:	Integración de componentes Enterprise Java Beans y Front End Angular para analizar el desempeño de acceso a datos
Código:	7697
Tipo de Publicación:	Artículo
Estado:	Publicado
Validación:	Para Actualización
Fecha de Publicación:	28/09/2023
Fecha de Registro:	18/01/2026
Fillación UTN:	Sí

Información de la Revista

Nombre de la Revista:	INNOVATION & DEVELOPMENT IN ENGINEERING AND
-----------------------	---

Figura 43 Autores de obra

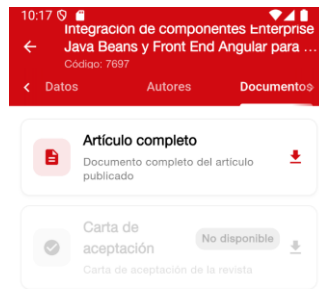
9:21 Integración de componentes Enterprise Java Beans y Front End Angular para ... Código: 7697

< Datos Autores Documento >

QUIÑA MERA JOSE ANTONIO
Cédula: 1002322384 Interno

REA PEÑAFIEL XAVIER MAURICIO
Cédula: 1002485744 Principal Interno

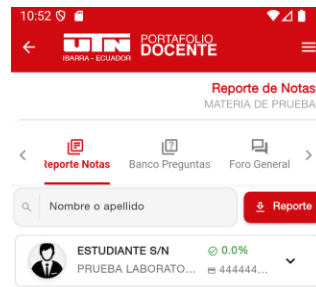
Figura 44 Documentos de obra



El siguiente módulo por trabajar del sprint fue el de foro online. Se implementó tanto el foro general como el foro de una actividad académica. Respecto al desarrollo del frontend, se trabajó en conjunto con el Est. Jordan Puruncajas, encargado de la Fase 2 del portafolio estudiante, para que ambos portafolios tengan las mismas interfaces.

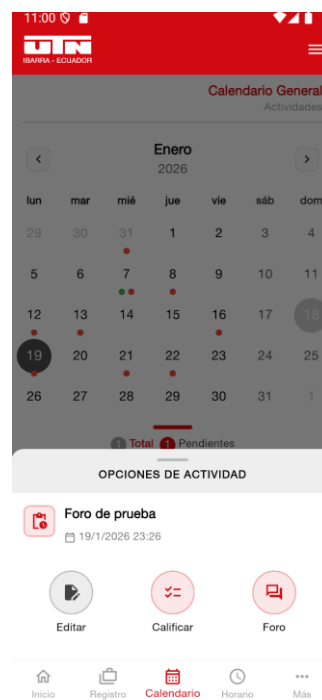
Como se observa en la **Figura 45**, el foro general es accesible desde el aula virtual de una materia, como una nueva pestaña.

Figura 45 Acceso a foro general



El foro de actividad está disponible al seleccionar una actividad, ya sea desde la lista de actividades del aula virtual o desde el calendario general, como una nueva opción.

Figura 46 Acceso a foro de actividad



En un foro online se visualizan los comentarios tanto de estudiantes como del docente. Cada comentario muestra su autor, con foto incluida, la fecha y si dicho comentario fue eliminado.

Figura 47 Comentarios de foro



Al crear o editar un comentario, se proporciona al docente un editor enriquecido con múltiples opciones, como negrita, sangría, tachado, sección de cita o código.

Figura 48 Participación en foro



El último módulo implementado en el sprint 2 fue el de control de bienes, en el cual el docente visualiza todos sus activos o bienes asignados. Estos pueden filtrarse por condición, rango de precio de compra y ordenarse por más antiguos o más recientes.

Figura 49 Lista de control de bienes

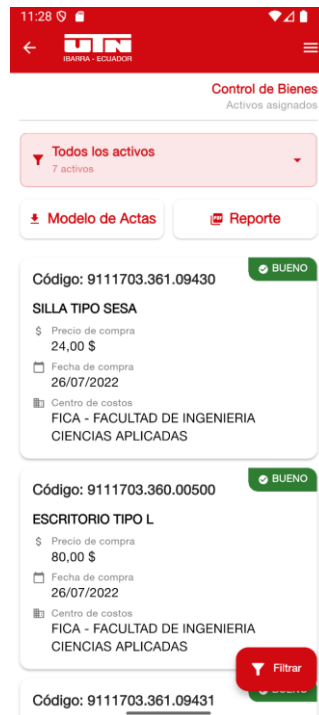


Figura 50 Filtrado de control de bienes

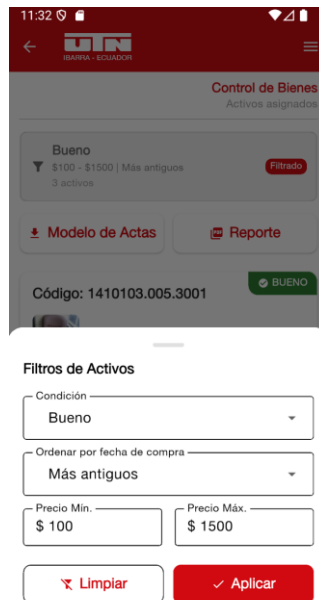
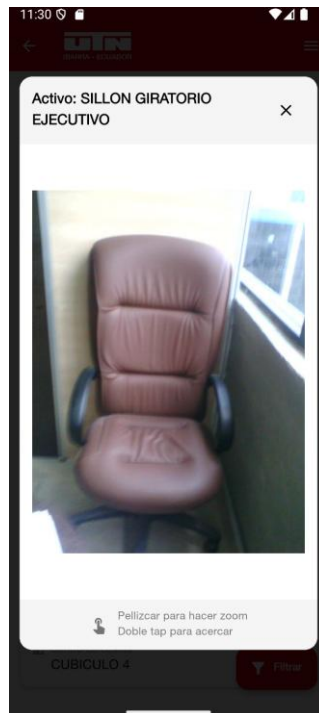


Figura 51 Visualización de fotografía de activo



También se puede generar un reporte PDF, ya sea de los activos o de bienes. El reporte generado es el mismo que se genera en el SIU.

Figura 52 Tipos de reporte de control de bienes

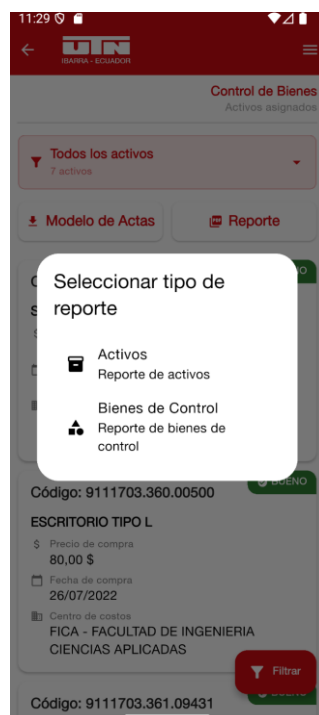


Figura 53 Reporte de control de bienes generado en PDF



2.2.3.3 Revisión del sprint 2

Tabla 28 Revisión de éxito por criterios de aceptación del sprint 2

Código	Criterios de Aceptación	Cumple
HU6	CA1: Al acceder a la sección de producción científica, puedo escoger entre todos los tipos de obras para acceder a su respectivo listado: artículos, libros, capítulos de libro, ponencias, y obras artísticas.	SI
	CA2: Cada obra en el listado debe mostrar datos básicos que comparten todos los tipos de obra como: título, tipo de obra, fecha de registro, fecha de publicación, estado, si tiene filiación con la UTN y su estado de validación.	SI
	CA3: Según el tipo de obra, cada una debe mostrar campos únicos correspondientes a su categoría, tales como base de	SI

	datos indexada, idioma, ciudad, país, tipo de obra artística, etc.	
HU7	CA1: Tras seleccionar una obra desde el listado previo, se abrirá una página con los datos generales, autores, y documentos de la obra.	SI
	CA2: En la sección de datos generales se mostrarán los siguientes campos, comunes a todos los tipos de obra: título, tipo de obra, fecha de registro, fecha de publicación, estado, filiación con la UTN, estado de validación, área de investigación, línea de investigación, campo amplio, campo específico, campo detallado, proyecto de investigación asociado y observaciones.	SI
	CA3: En la sección de datos generales, según el tipo de obra, cada una debe mostrar campos únicos correspondientes a su categoría, tales como: base de datos indexada, editorial, link, keywords, número de páginas, idioma, ciudad, país, tipo de obra artística, etc.	SI
	CA4: En la sección de autores, se mostrará un listado con los siguientes datos: nombres, cédula, si es autor externo o principal.	SI
	CA5: En la sección de documentos, según el tipo de obra, cada una debe mostrar documentos específicos correspondientes a su categoría disponibles para descargar, tales como: documento completo, documento de publicación, informes 1 y 2 de pares, certificado de ponencia, documento de SENADI, etc.	SI
HU8	CA1: Tras acceder al aula virtual de un curso, debe estar disponible el apartado de foro general.	SI
	CA2: Si existen respuestas, deben mostrarse las diferentes participaciones tanto de docentes como de estudiantes, incluyendo datos como descripción, fecha de creación y autor.	SI

	CA3: Puedo agregar nuevas respuestas; el texto de la descripción debe permitir formato enriquecido.	SI
	CA4: Puedo editar mis respuestas.	SI
	CA5: Puedo eliminar mis respuestas; al seleccionar la opción de eliminar, debe mostrarse un mensaje de confirmación antes de realizar la eliminación.	SI
HU9	CA1: Una vez dentro del aula virtual de un curso, al seleccionar una actividad, si es de tipo foro dentro de la plataforma institucional, debe estar disponible la opción para acceder al foro.	SI
	CA2: Si existen respuestas, deben mostrarse las diferentes participaciones tanto de docentes como de estudiantes, incluyendo datos como descripción, fecha de creación y autor.	SI
	CA3: Puedo agregar nuevas respuestas; el texto de la descripción debe permitir formato enriquecido.	SI
	CA4: Puedo editar mis respuestas.	SI
	CA5: Puedo eliminar mis respuestas; al seleccionar la opción de eliminar, debe mostrarse un mensaje de confirmación antes de realizar la eliminación.	SI
HU10	CA1: La lista de activos debe incluir los siguientes datos: código, nombre, observación, precio de compra, fecha de compra, fotografía y condición actual.	SI
	CA2: Dentro de la sección de control de bienes, la app debe permitir la descarga del formato de Word "Modelo de Actas".	SI
	CA3: Se puede generar y visualizar el reporte de control de bienes, según el tipo escogido: activos o bienes.	SI
	CA4: Se puede filtrar el listado de activos por condición, rango de precio de compra (mínimo y máximo), así como ordenarlo ascendente y descendentemente por la fecha de compra.	SI

2.2.3.4 Retrospectiva del sprint 2

Este sprint destacó por la colaboración efectiva con el Est. Jordan Puruncajas en el desarrollo de los módulos de foro, logrando interfaces consistentes entre ambos portafolios. El módulo de producción científica requirió un manejo dinámico de campos según el tipo de obra, lo que implicó un diseño cuidadoso de componentes reutilizables. La implementación del editor enriquecido para los foros presentó desafíos técnicos que fueron superados mediante el estudio de librerías especializadas. Se evidenció la importancia de mantener una comunicación constante entre desarrolladores para evitar duplicación de esfuerzos y garantizar coherencia en la experiencia de usuario.

2.2.4 Sprint 3

2.2.4.1 Planificación del sprint 3

Tabla 29 Sprint backlog 3

Código	Nombre	Tarea	Horas
HU11	Lista de preguntas	Desarrollo de endpoint GET	3
		Desarrollo de pruebas unitarias	2
		Desarrollo de interfaz	5
HU12	Detalle de pregunta	Desarrollo de endpoint GET	2
		Desarrollo de pruebas unitarias	1
		Desarrollo de interfaz	5
HU13	Crear pregunta	Desarrollo de endpoint POST	3
		Desarrollo de pruebas unitarias	3
		Desarrollo de interfaz	7
HU14	Editar pregunta	Desarrollo de endpoint UPDATE	3
		Desarrollo de pruebas unitarias	3
		Desarrollo de interfaz	4
HU15	Eliminar pregunta	Desarrollo de endpoint DELETE	1
		Desarrollo de pruebas unitarias	1
		Desarrollo de interfaz	1
HU16	Importar preguntas	Desarrollo de endpoint GET de preguntas a importar	1

Desarrollo de endpoint POST de importar preguntas seleccionadas	3
Desarrollo de endpoint POST de importar de un contenido	2
Desarrollo de endpoint POST de importar preguntas de una materia	2
Desarrollo de pruebas unitarias	4
Desarrollo de interfaz	8
TOTAL	64

2.2.4.2 Incremento del sprint 3

El Sprint 3 se enfoca en la implementación del banco de preguntas de una materia, un espacio en el que los docentes pueden visualizar, agregar, importar, editar y eliminar preguntas asociadas a una materia específica. Estas preguntas pueden ser utilizadas posteriormente en evaluaciones online.

El banco de preguntas es accesible a través de una nueva pestaña dentro del aula virtual de la materia. Este módulo presenta un listado de preguntas, desde el cual el docente puede seleccionar cada una para visualizar su detalle, editarla o eliminarla. Adicionalmente, desde esta misma sección es posible agregar nuevas preguntas, aplicar filtros y realizar la importación de preguntas desde otra aula virtual.

Figura 54 Lista de preguntas

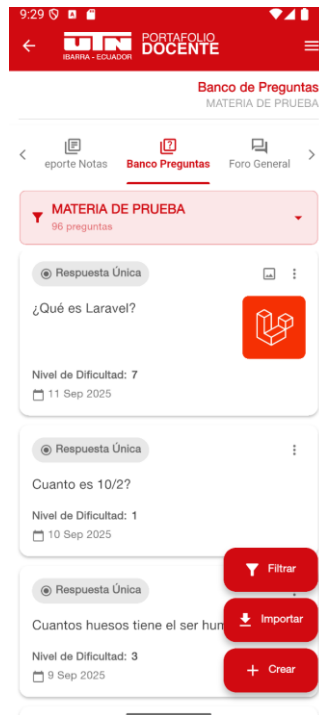


Figura 55 Filtrado de preguntas

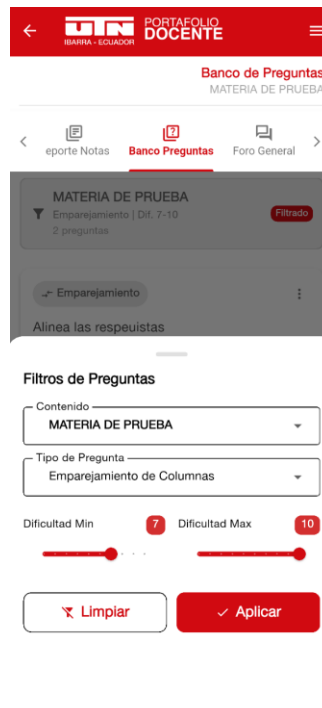


Figura 56 Opciones de pregunta

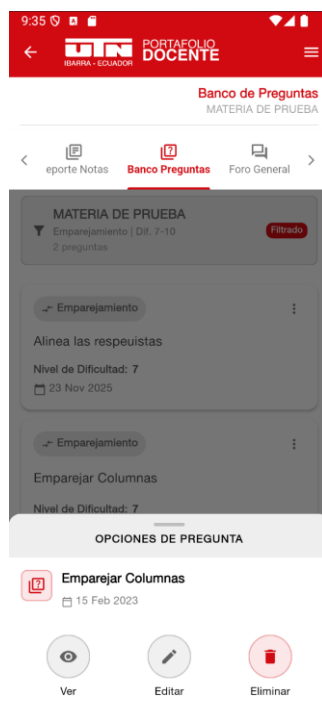
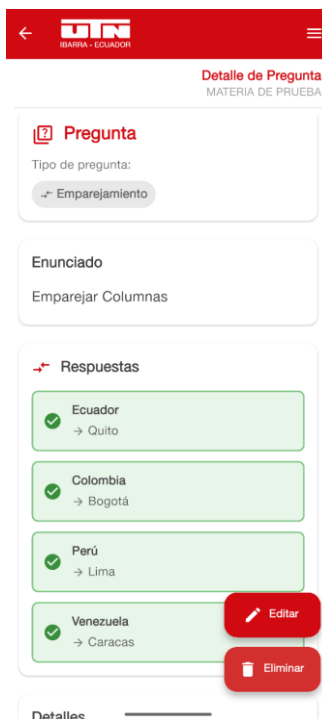


Figura 57 Detalle de pregunta



Tanto al agregar como al editar una pregunta, se carga un formulario que permite modificar todos los campos de la pregunta. Además, en el caso de preguntas que no sean de tipo ensayo, el formulario permite agregar y gestionar las alternativas de respuesta.

Figura 58 Formulario de pregunta sección 1

9:39 UNISA - ECUADOR

Editar Pregunta
MATERIA DE PRUEBA

Datos de la Pregunta

Tipo de Pregunta
Emparejamiento de Columnas

Contenido
UNIDAD 1

Nivel de Dificultad
7 Dificil

1 10

B I U

Emparejar Columnas

Imagen Adjunta (Opcional)
(Máx. 5MB)

X Cancelar Guardar

Figura 59 Formulario de pregunta sección 2

9:39 UNBARRA - ECUADOR

Editar Pregunta
MATERIA DE PRUEBA

Imagen Adjunta (Opcional)
(Máx. 5MB)

Seleccionar una imagen

Alternativas de Respuesta

Seleccionar para eliminar

Respuestas + Agregar

1

Columna A1
Ecuador

Columna B1 (Emparejamiento)
Quito

Cancelar Guardar

En el apartado de importar preguntas, el docente selecciona diferentes parámetros para importar preguntas desde el aula virtual de otra materia hacia el aula virtual actual. Es posible seleccionar preguntas individuales, importar todas las preguntas asociadas a un contenido específico o importar todas de las preguntas de la materia.

Figura 60 Formulario de importar preguntas sección 1

10:24

UN
BARBA - ECUADOR

Importar Preguntas
MATERIA DE PRUEBA 2

Ciclo Académico
OCT2021-OCT2099

Materia Origen
MATERIA DE PRUEBA A(Carrera Prueba)

Contenido Origen
1.1 CONTENIDO 1.1

Contenido Destino
1.1 Proyecto basado en un marco de trabajo ágil apoyado en herramientas de inteligencia artificial

Seleccionar Preguntas
Selección individual

Seleccione cómo desea importar:

☒ Seleccionar preguntas individuales
Elija específicamente qué preguntas importar de este contenido

☐ Importar todo el contenido
Importar todas las preguntas del contenido origen

Total de preguntas: 11

Cancelar Importar

Figura 61 Formulario de importar preguntas sección 2

UN
BARBA - ECUADOR

Importar Preguntas
MATERIA DE PRUEBA 2

Total de preguntas: 11
Seleccionadas: 5

Deseleccionar todas

☒ Nivel 7
Cuanto tiempo aproximado de vida tenían los cavernícolas
Respuesta Única
20 Ene 2026 09:06

☐ Nivel 1
All in *DUE TIME* - KWS
Respuesta Única
18 Ene 2026 10:41

☐ Nivel 7
Crear imagen comp
Ensayo o Respuesta Textual
15 Ene 2026 06:41

☒ Nivel 3
Empareje Nombre - Apellido

Cancelar Importar

2.2.4.3 Revisión del sprint 3

Tabla 30 Revisión de éxito por criterios de aceptación del sprint 3

Código	Criterios de Aceptación	Cumple
HU11	CA1: Las preguntas deben incluir datos como: enunciado, tipo de pregunta, nivel de dificultad, imagen y fecha de creación.	SI
	CA2: Cada pregunta debe proporcionar las siguientes opciones: ver(detalle), editar y eliminar.	SI
	CA3: Desde la página de listado de preguntas, deben estar disponible el acceso a la funcionalidad de crear e importar preguntas.	SI
	CA4: Se puede filtrar el listado de preguntas por contenido, tipo, rango de dificultad.	SI
HU12	CA1: Se deben incluir los siguientes datos de la pregunta: enunciado, tipo de pregunta, nivel de dificultad, imagen y fecha de creación.	SI
	CA2: Se deben mostrar las alternativas de respuesta de la pregunta cuando esta no sea de tipo ensayo. Para preguntas de respuesta única u opción múltiple, se deben mostrar las alternativas con sus respectivos enunciados, indicando claramente la(s) respuesta(s) correcta(s). Para preguntas de emparejamiento de columnas, se deben mostrar los enunciados de ambas columnas junto con sus correspondencias.	SI
	CA3: Desde la página de detalle de la pregunta, debe ser posible editarla o eliminarla.	SI
HU13	CA1: El formulario para crear una pregunta debe contar con los siguientes campos obligatorios: enunciado (con editor enriquecido), contenido, tipo, y nivel de dificultad. El formulario debe permitir ingresar una imagen de forma opcional.	SI

HU14	CA2: No se puede agregar una pregunta si el ciclo académico de la materia es inactivo.	SI
	CA3: Según el tipo de pregunta seleccionado, el formulario debe mostrar dinámicamente los campos. Si es respuesta única u opción múltiple: ingreso de alternativas y selección de la(s) correcta(s). Si es emparejamiento de columnas, ingreso de enunciados para ambas columnas y su correspondencia. Y si es ensayo, no se deben mostrar campos de alternativas.	SI
	CA3: La app debe validar que todos los campos obligatorios estén completos antes de permitir el guardado. Si la pregunta a crear no es tipo ensayo, se deben realizar validaciones antes de guardar la pregunta, como verificar que en respuesta única u opción múltiple al menos exista una respuesta correcta.	SI
	CA5: Al guardar la pregunta correctamente, la app debe registrar la fecha de creación y redirigir a la página de listado de preguntas.	SI
	CA1: Desde la página de detalle de la pregunta y en el listado de preguntas, debe existir una opción para editar una pregunta específica.	SI
	CA2: No se puede editar una pregunta si el ciclo académico de la materia es inactivo.	SI
	CA3: Se debe mostrar un formulario precargado con la información actual de la pregunta y alternativas de respuesta.	SI
	CA4: El formulario debe permitir modificar los mismos campos definidos para la creación de la pregunta y alternativas de respuesta. Las validaciones deben ser las mismas que en la creación de la pregunta.	SI
	CA5: Al guardar los cambios, la información de la pregunta debe actualizarse y reflejarse en la página de listado de preguntas.	SI

HU15	CA1: Desde la página de listado de preguntas y en la de detalle, debe existir una opción para eliminar.	SI
	CA2: No se puede eliminar una pregunta si el ciclo académico de la materia es inactivo o si está ya pertenece al cuestionario de una evaluación online.	SI
	CA3: Al seleccionar la opción de eliminar, la app debe solicitar una confirmación explícita al docente.	SI
	CA4: Si se confirma la eliminación, la pregunta debe marcarse como eliminada (eliminación lógica) y no debe ser visible en el banco de preguntas.	SI
HU16	CA1: Al seleccionar “Importar” desde la página de lista de preguntas, se visualiza el flujo de importación por pasos con selectores de ciclo, materia y contenido.	SI
	CA2: No se puede importar preguntas si el ciclo académico de la materia es inactivo.	SI
	CA3: En el paso de origen, el docente elige un contenido en base al ciclo y materia seleccionados. En el paso de destino, el docente elige el contenido de su materia actual en donde se importarán las preguntas.	SI
	CA4: En el paso final, el docente elige el modo: “individual” (puede ver preguntas, seleccionar/deseleccionar, seleccionar todas las no importadas) o “todo” (ve un resumen de cuántas se importarán).	SI
	CA5: La acción de importar solo se habilita en el último paso cuando las validaciones se cumplen (en “individual”, hay preguntas seleccionadas). Al finalizar la importación, la app redirige al listado de preguntas en donde se reflejan las preguntas recién importadas.	SI

2.2.4.4 Retrospectiva del sprint 3

El desarrollo del banco de preguntas fue desafiante debido a la necesidad de manejar diferentes tipos de preguntas con sus respectivas alternativas de respuesta de forma dinámica. La implementación del flujo de importación de preguntas requirió un análisis detallado de los

requisitos para garantizar validaciones apropiadas en cada paso. El uso de formularios dinámicos que se adaptan según el tipo de pregunta seleccionado demostró ser una solución efectiva que mejoró significativamente la experiencia del usuario.

2.2.5 Sprint 4

2.2.5.1 Planificación del sprint 4

Tabla 31 Sprint backlog 4

Código	Nombre	Tarea	Horas
HU17	Agregar preguntas a evaluación	Desarrollo de endpoint GET de preguntas ya agregadas	2
		Desarrollo de endpoint GET de preguntas disponibles	1
		Desarrollo de endpoint POST de agregar preguntas seleccionadas	2
		Desarrollo de endpoint DELETE de eliminar preguntas seleccionadas	1
		Desarrollo de endpoint DELETE de eliminar todas las preguntas de una evaluación	1
		Desarrollo de pruebas unitarias	2
		Desarrollo de interfaz	4
H18	Sortear preguntas de evaluación	Desarrollo de endpoint GET de número de preguntas disponibles	1
		Desarrollo de endpoint POST de sortear preguntas	1
		Desarrollo de endpoint GET de preguntas sorteadas por cada estudiante	2
		Desarrollo de pruebas unitarias	2
		Desarrollo de interfaz	3

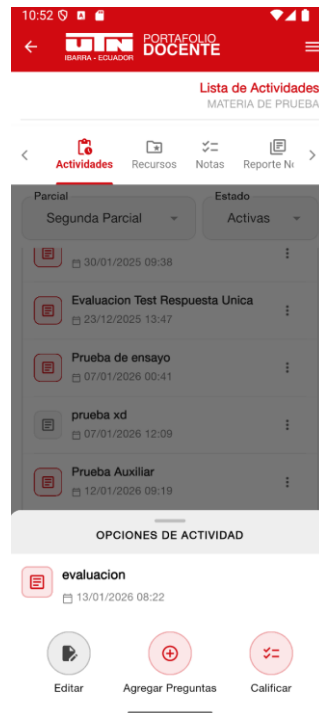
HU19	Revisión de evaluación online	Desarrollo de endpoint GET de cuestionario rendido por un estudiante	4
		Desarrollo de pruebas unitarias	1
		Desarrollo de interfaz	5
HU20	Calificación de preguntas de ensayo	Desarrollo de endpoint POST de calificar una pregunta ensayo	1.5
		Desarrollo de pruebas unitarias	1.5
		Desarrollo de interfaz	1
TOTAL			36

2.2.5.2 Incremento del sprint 4

Este sprint abarca todas las funcionalidades relacionadas con las evaluaciones online.

Una vez creada una actividad académica de tipo evaluación online dentro de la plataforma institucional, y siempre que el tipo de evaluación sea manual, al seleccionar dicha actividad creada se habilita la opción para agregar preguntas como se observa en la **Figura 62**.

Figura 62 Opción de agregar preguntas

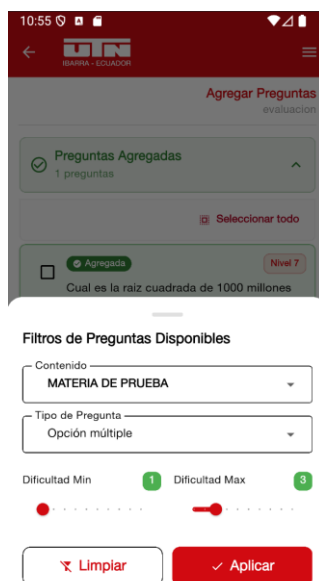


El apartado de agregar preguntas se compone de dos listados: el de preguntas ya agregadas a la evaluación y el de preguntas disponibles en el banco de preguntas. Adicionalmente, es posible aplicar filtros sobre el listado de preguntas disponibles y se dispone de un acceso directo al formulario de creación de preguntas desde esta misma sección.

Figura 63 Listados de preguntas agregadas y disponibles de evaluación



Figura 64 Filtrado de preguntas disponibles



Desde el listado de preguntas agregadas se pueden quitar preguntas de la evaluación, mientras que desde el listado de preguntas disponibles se agregan preguntas a dicha evaluación.

Figura 65 Quitar preguntas de evaluación

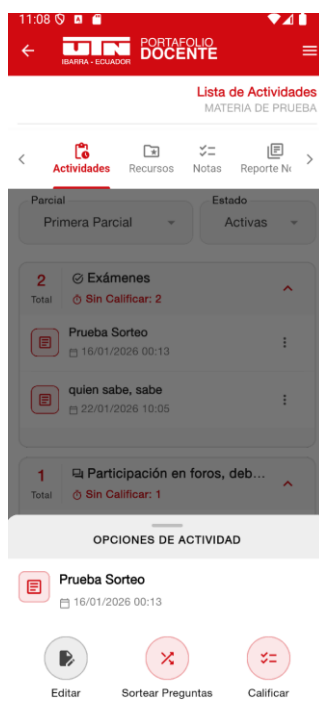


Figura 66 Agregar preguntas a evaluación



La siguiente funcionalidad implementada fue el sorteo de preguntas de una evaluación online de tipo automática. La misma también es accesible como una nueva opción al seleccionar la actividad académica, como se observa en la **Figura 67**.

Figura 67 Opción de sortear preguntas



Al acceder por primera vez a esta nueva página, se presenta el apartado de configuración del sorteo. Una vez que el docente ha definido los parámetros correspondientes, se muestra el número de preguntas disponibles en el banco de preguntas. Posteriormente, el docente debe indicar la cantidad de preguntas a sortear, así como especificar si desea que se asignen preguntas diferentes a cada estudiante.

Figura 68 Configuración del sorteo

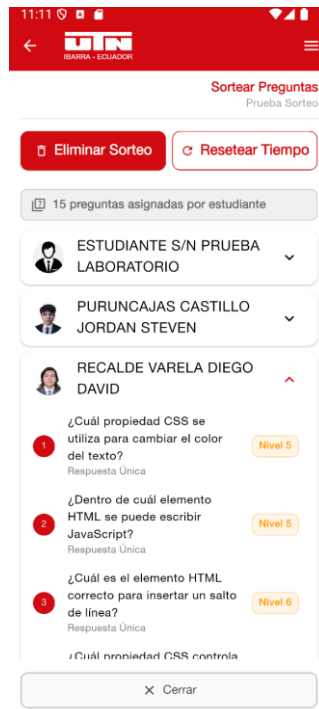
The screenshot shows the 'Configuración del Sorteo' (Lottery Configuration) screen in the UNIMSA ECUADOR app. The screen is titled 'Sortear Preguntas' (Draw Questions) and 'Prueba Sorteo' (Lottery Test). The configuration options include:

- Contenido:** A dropdown menu set to 'MATERIA DE PRUEBA'.
- Dificultad Mínima:** A slider set to 5.
- Dificultad Máxima:** A slider set to 6.
- Preguntas disponibles:** 23.
- Número de preguntas:** A text input field set to 15.
- Cuestionario diferente:** A toggle switch set to 'On' (red).

At the bottom, there are two buttons: 'Cancelar' (Cancel) and 'Sortear' (Draw).

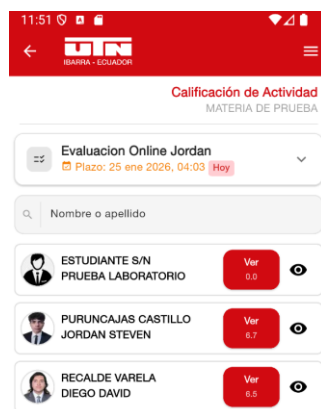
Una vez finalizado el sorteo, se presenta el listado de estudiantes de la materia junto con las preguntas asignadas en la evaluación en línea. Adicionalmente, desde esta sección es posible eliminar todas las preguntas generadas por el sorteo para realizarlo nuevamente, así como utilizar la opción de restablecer el tiempo de la evaluación.

Figura 69 Preguntas sorteadas a cada estudiante



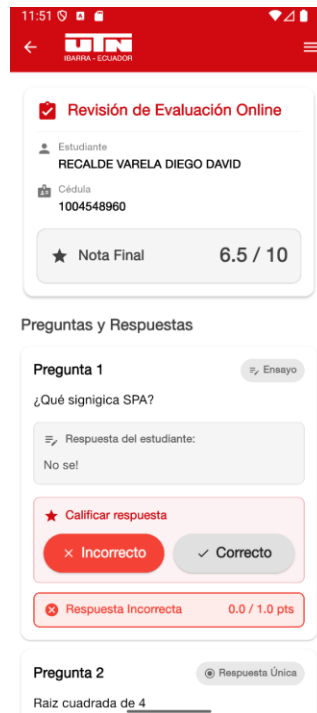
La última funcionalidad abordada en este sprint corresponde a la revisión y calificación de evaluaciones en línea. Desde el apartado de calificación de la actividad, cuando esta corresponde a una evaluación en línea, cada estudiante dispone de un botón “Ver”, a través del cual se puede acceder a la revisión del intento realizado.

Figura 70 Acceso a revisión de evaluación



Desde este nuevo apartado es posible visualizar la calificación y el intento de evaluación de un estudiante, incluyendo todas las preguntas y las respuestas proporcionadas. La calificación de todas las preguntas, con excepción de las de tipo ensayo, se realiza de manera automática. En el caso de las preguntas de tipo ensayo, el docente debe calificarlas manualmente desde esta misma sección, marcándolas como correctas o incorrectas.

Figura 71 Revisión de evaluación online



2.2.5.3 Revisión del sprint 4

Tabla 32 Revisión de éxito por criterios de aceptación del sprint 4

Código	Criterios de Aceptación	Cumple
HU17	CA1: Al seleccionar una actividad desde el listado de actividades del aula virtual o desde el calendario general, si la actividad es de tipo evaluación online y la evaluación es de tipo manual, la app debe mostrar la opción "Agregar Preguntas".	SI
	CA2: Al acceder a la opción "Agregar Preguntas", la app debe mostrar una página dividida en dos secciones.	SI
	CA3: En "preguntas agregadas" debe visualizarse un listado de las preguntas que ya forman parte del cuestionario, desde el cual se pueden seleccionar una o más preguntas para eliminarlas de la evaluación.	SI
	CA4: En "preguntas disponibles" debe visualizarse un listado de preguntas disponibles en el banco de preguntas, desde el	SI

	cual se pueden seleccionar una o más preguntas para agregarlas a la evaluación.	
	CA5: Deben estar disponibles las acciones de eliminar todas las preguntas, así como restablecer el tiempo de evaluación de los estudiantes.	SI
HU18	CA1: Al seleccionar una actividad desde el listado de actividades del aula virtual o desde el calendario general, si la actividad es de tipo evaluación online y la evaluación es de tipo automática, la app debe mostrar la opción "Sortear Preguntas".	SI
	CA2: Al acceder a la página de sorteo, cuando aún no se haya realizado el mismo, se debe mostrar un formulario de configuración que permita definir el contenido, la dificultad mínima y máxima, la opción de generar un cuestionario diferente para cada estudiante y el número de preguntas; en función de estos parámetros, el sistema debe mostrar el total de preguntas disponibles en el banco de preguntas.	SI
	CA3: La acción de sortear solo se permite si se han configurado todos los parámetros correctamente y se ha escrito un número de preguntas válido.	SI
	CA4: Finalizado el sorteo, se muestra la vista de estudiantes con el total de preguntas asignadas por estudiante; cada estudiante despliega sus preguntas sorteadas con enunciado, tipo y nivel de dificultad.	SI
	CA5: Deben estar disponibles las acciones de eliminar todas las preguntas, así como restablecer el tiempo de evaluación de los estudiantes.	SI
HU19	CA1: Al acceder al apartado de calificaciones de una actividad de tipo evaluación online, se debe permitir revisar el intento de evaluación de todos los estudiantes.	SI
	CA2: Deben visualizarse todas las preguntas de la evaluación con: enunciado, nota, tipo, nivel de dificultad y alternativas de respuesta si es no es de tipo ensayo.	SI

HU20	CA3: Para preguntas de tipo ensayo, debe indicarse que requieren calificación manual.	SI
	CA4: Debe mostrarse el puntaje total obtenido por el estudiante en la evaluación.	SI
	CA1: Se debe permitir al docente calificar únicamente las preguntas de tipo ensayo del intento de evaluación de un estudiante.	SI
	CA2: Cada pregunta de tipo ensayo solo debe poder calificarse con los valores correcta (1) o incorrecta (0).	SI
	CA3: No se puede calificar preguntas si el ciclo académico de la materia es inactivo.	SI
	CA4: Al registrar la calificación de una pregunta de tipo ensayo, la app debe recalcular automáticamente la nota total de la evaluación.	SI

2.2.5.4 Retrospectiva del sprint 4

Este sprint final integró exitosamente todas las funcionalidades de evaluaciones online, culminando el desarrollo de la Fase 2 del portafolio docente. La implementación de la revisión y calificación de evaluaciones demostró la importancia de una arquitectura bien diseñada desde el Sprint 3, facilitando la reutilización de componentes del banco de preguntas. El equipo Scrum validó satisfactoriamente todas las funcionalidades, confirmando que el producto final cumple con los lineamientos institucionales y está listo para la etapa de pruebas con usuarios reales.

2.3 Etapa de post-juego

Con el apoyo del Ing. Franklin Enríquez del DDTI, se realizó el despliegue del microservicio del portafolio docente en el entorno de desarrollo. Siguiendo el mismo enfoque de la Fase 1, se mantuvo la estrategia de despliegue continuo mediante la implementación de un workflow automático con Git, Docker y GitHub Actions. Este workflow permite que, al realizar un commit y push a la rama principal del repositorio del portafolio docente, el servidor que aloja los microservicios se actualice automáticamente con los cambios recientes. El uso de Docker garantizó la portabilidad y el aislamiento del microservicio, facilitando su ejecución en diferentes entornos sin conflictos de dependencias. Esta infraestructura de integración continua optimiza el

proceso de desarrollo al reducir significativamente los tiempos de despliegue y minimizar los errores humanos durante la actualización del sistema en producción.

El entorno de desarrollo permitió realizar tanto pruebas unitarias como de integración en un ambiente casi idéntico al de producción, lo que facilitó la evaluación de aspectos críticos como rendimiento, estabilidad e interoperabilidad del microservicio del portafolio docente con los demás microservicios y el API Gateway. Se validaron exitosamente todas las funcionalidades implementadas: visualización de proyectos de investigación, obras de producción científica, pagos y control de bienes, así como la gestión completa de foros, evaluaciones online y banco de preguntas. El trabajo en un entorno real también permitió identificar y corregir errores inesperados relacionados con la integración de componentes y optimizar endpoints con consultas SQL ineficientes.

Como resultado de la etapa de post-juego, se logró un producto estable y funcional que cumple con todos los requisitos establecidos en el Product Backlog. Las 20 historias de usuario implementadas a lo largo de todos los sprints fueron validadas satisfactoriamente por el Product Owner y los stakeholders del DDTI, quienes confirmaron que las funcionalidades desarrolladas se alinean con los lineamientos institucionales de la UTN y están listas para su eventual migración al entorno de producción. El microservicio del portafolio docente demostró una integración efectiva con la arquitectura existente de UTN-Móvil, manteniendo tiempos de respuesta aceptables y garantizando la consistencia de datos con el SIIU.

III VALIDACIÓN DE RESULTADOS

Tras completar el desarrollo de la Fase 2 del portafolio docente en UTN-Móvil utilizando la metodología ágil Scrum, se procede a evaluar la calidad del producto obtenido desde la perspectiva del usuario final. Este capítulo documenta el proceso de validación de la calidad en uso, basado en la norma ISO/IEC 25022, mediante el cual se evalúan las características de eficacia, eficiencia y satisfacción del sistema. La evaluación se realizó con una muestra de 20 docentes de las cinco facultades de la Universidad Técnica del Norte, aplicando un taller práctico y la encuesta SUS. Los resultados obtenidos evidencian que el sistema alcanza un 88.28% de calidad en uso, superando los requisitos establecidos y confirmando su calidad para apoyar la gestión académica docente.

3.1 Definición del modelo de calidad en uso

Una vez finalizada la fase de desarrollo de la Fase 2 del portafolio docente, resulta fundamental realizar la evaluación de este aplicativo móvil. En concordancia con los objetivos definidos del trabajo de titulación, se evaluó la calidad en uso conforme a la norma ISO/IEC 25022. Para este proceso se utilizó el mismo modelo de calidad en uso empleado en la evaluación de la Fase 1 [3]; por lo tanto, además de las características previamente definidas, se tienen en cuenta subcaracterísticas complementarias, cada una con una con un porcentaje de impacto específico. Todas las características y subcaracterísticas se observan en la **Tabla 33**.

Tabla 33 Modelo de Calidad en Uso

Modelo de Calidad en Uso			
Característica	Sub Característica	Peso de Característica	Peso de Sub Característica
Eficacia	Tareas completadas	42%	16%
	Objetivos completados		16%
	Tareas sin errores		10%
Eficiencia	Tiempo de tareas	32%	16%
	Eficiencia de tiempo		16%
Satisfacción	Utilidad	26%	10%
	Confianza		8%
	Comodidad		8%

3.2 Medición de la calidad en uso

Tomando como base el modelo de calidad en uso previamente definido, se utilizaron los mismos instrumentos de evaluación empleados en la Fase 1: un taller práctico y la encuesta SUS. La característica de satisfacción se determinó a través de la encuesta SUS, mientras que las demás características de eficacia y eficiencia se evaluaron mediante al taller práctico.

3.2.1 Muestra poblacional

Para la evaluación de la Fase 2 se contó con la participación de 20 docentes pertenecientes a las cinco facultades de la universidad: FECYT (4), FACA E (5), FCCSS (4), FICA (4) y FICAYA (3). Gracias a la gestión del Ing. Jorge Caraguay, director del DDTI, se solicitó a los decanos de cada facultad la designación de entre 3 y 5 docentes, priorizando a aquellos que ya utilizan la aplicación móvil UTN, con el fin de participar en el proceso de evaluación.

De este grupo, se identificó a cinco docentes como “usuarios expertos”, en función de su uso del portafolio docente en UTN-Móvil (Fase 1) y del portafolio docente web del SIIU. Así como se hizo en la muestra de la Fase 1, se identificó un docente por facultad con el fin de no sesgar los resultados. Dichos docentes expertos son los que tienen el ID 5, 8, 18, 19 y 21, como se observa en el **Anexo C y D**.

Al igual que en la fase anterior, se excluyó a los docentes que desempeñan funciones en el área de vinculación con la sociedad, debido a que el alcance de la Fase 2 no comprende funcionalidades relacionadas a vinculación.

3.2.2 Taller práctico

El taller práctico, con una duración aproximada de 15 minutos, se compone de ocho actividades clave que abarcan las funcionalidades más importantes y utilizadas del SIIU. Cada actividad se representa mediante un objetivo, el cual se concreta en una tarea específica. Una vez que el docente accede al servicio de portafolio docente en el aplicativo móvil, el taller inicia con los objetivos y tareas detallados en la **Tabla 34**.

Tabla 34 Diseño del taller práctico

Nro.	Objetivo	Tarea
1	Ver pagos	Desde el menú principal, seleccionar la opción Más y luego Pagos: - Seleccionar un pago para expandirlo y ver sus conceptos. - Seleccionar Factura para visualizar el documento PDF.
2	Ver control de bienes	Seleccionar el menú Más y luego Control de Bienes: -Seleccionar Reporte. -Seleccionar Reporte de activos.
3	Ver proyecto de investigación	Seleccionar el menú Más y luego Proyectos de Investigación: -Seleccionar 'Mis Proyectos' -Seleccionar el primer proyecto del listado -Escoger la pestaña de Equipo y visualizar el listado de miembros.

4	Ver obra de producción científica	Ingresar al menú Más y luego Producción Científica: -Seleccionar Artículos (En caso de no tener escoger otro tipo de obra) -Seleccionar primera obra del listado -Visualizar cualquiera de las pestañas del detalle de obra
5	Comentar en foro general	Ingresar al Registro Académico, luego al Aula Virtual de una materia: -Seleccionar Foro General (la última pestaña de la barra superior). -Seleccionar botón de crear (esquina inferior derecha). -Escribir comentario -Enviar comentario
6	Crear evaluación online	Dentro del Aula Virtual de la materia, ir a la pestaña Actividades (primera pestaña de la barra superior): -Seleccionar el botón Crear -Llenar título, instrucciones y contenido -Seleccionar Primera Parcial -En política de evaluación escoger Pruebas (30%) -En tipo de actividad escoger Evaluación Online (Dentro de Plataforma Institucional) -En configuración de evaluación online escoger Manual -Establecer límite de tiempo de evaluación -Crear actividad
7	Crear 3 preguntas	Desde la pestaña Actividades del Aula virtual, seleccionar la nueva actividad creada en la sección de Pruebas: -Escoger la opción Agregar Preguntas. -Seleccionar el botón Crear. -Escoger como tipo de pregunta Respuesta Única. -Seleccionar cualquier contenido y nivel de dificultad. -Escribir enunciado. -Escribir alternativas de respuesta, marcar al menos una como correcta. -Guardar pregunta.

8	Agregar preguntas a evaluación online	-Repetir proceso y crear 2 preguntas más. Dentro de la página Agregar Preguntas, expandir la sección de Preguntas Disponibles: -Seleccionar las tres preguntas creadas recientemente -Seleccionar acción de Agregar (debajo del título Preguntas Disponibles) -Confirmar acción de agregar 3 preguntas -Visualizar preguntas agregadas en sección Preguntas Agregadas
---	---------------------------------------	--

3.2.3 Encuesta SUS

Para evaluar la usabilidad del aplicativo móvil, se aplicó una encuesta en Microsoft Forms basada en el cuestionario SUS (System Usability Scale), compuesto por 10 ítems. Las respuestas se recolectaron mediante una escala Likert de cinco puntos, donde el valor 1 representa un "Totalmente en desacuerdo" y el 5 un "Totalmente de acuerdo", tal como se detalla en la Tabla 35.

Tabla 35 Preguntas encuesta SUS

Código	Pregunta
P1	¿Cree usted que usaría este sistema con frecuencia?
P2	¿Cree que el sistema es innecesariamente complejo?
P3	¿Cree que el sistema fue fácil de usar?
P4	¿Cree usted que necesitaría ayuda de un experto con conocimientos técnicos para utilizar este sistema?
P5	¿Cree que las diferentes funcionalidades del sistema están bien integradas?
P6	¿Cree que el sistema es inconsistente?
P7	¿Cree que la mayoría de la gente aprendería a usar este sistema muy rápidamente?
P8	¿Cree que el sistema es difícil de usar?
P9	¿Se sintió confiado al usar el sistema?

P10	¿Cree que se necesitan conocimientos técnicos avanzados antes de usar el sistema?
-----	---

Las preguntas de la encuesta SUS se agrupan en dos dimensiones de evaluación: utilidad y comodidad. La dimensión de utilidad, integrada por las preguntas P1, P6 y P9, permite evaluar el grado en que el sistema resulta funcional y confiable para el docente, considerando aspectos como la frecuencia de uso, la consistencia en su comportamiento y la confianza generada durante la interacción. Mientras que, la dimensión de comodidad, conformada por las preguntas P2, P3, P4, P5, P7, P8 y P10, evalúa la facilidad de uso del sistema y la experiencia del usuario, abordando criterios como la percepción de complejidad, la facilidad de aprendizaje, la necesidad de soporte técnico, la integración de funcionalidades, la dificultad de uso y el requerimiento de conocimientos previos.

3.2.4 Fiabilidad del taller práctico

Debido a que la totalidad de los docentes participantes completó satisfactoriamente todas las tareas propuestas en el taller práctico (20 de 20 participantes en las 8 tareas), y no se registraron errores durante su ejecución, no fue posible calcular indicadores estadísticos de fiabilidad, como el coeficiente alfa de Cronbach, ni aplicar pruebas de normalidad como Kolmogorov-Smirnov, debido a la ausencia de variabilidad en los resultados. Este comportamiento evidencia un alto nivel de facilidad de uso y comprensión del sistema por parte de los docentes.

3.2.5 Fiabilidad de la encuesta SUS

A partir de los datos tabulados de los docentes encuestados, presentados en el **Anexo D**, se realizó la validación de la fiabilidad del instrumento mediante el coeficiente estadístico alfa de Cronbach en IBM SPSS. El valor del alfa de Cronbach se interpreta conforme a los criterios establecidos: resultado superior a 0.9 se considera excelente, entre 0.8 y 0.9 bueno, valor igual o superior a 0.7 aceptable, e inferior a 0.6 indican una fiabilidad reducida. En adición, teniendo en cuenta que algunas preguntas tienen orientación negativa, se aplicó un proceso de recodificación a los ítems P2, P4, P6, P8 y P10, con el propósito de unificar el sentido de las respuestas y asegurar que valores más altos representen una mejor percepción del sistema.

Figura 72 Alfa de Cronbach encuesta SUS

Estadísticas de fiabilidad		
Alfa de Cronbach	Alfa de Cronbach basada en elementos estandarizados	N de elementos
.720	.752	10

Fuente: Propia

Los resultados obtenidos del análisis de fiabilidad del instrumento, mediante el coeficiente alfa de Cronbach, evidencian un valor de 0.720, mientras que el alfa basado en elementos estandarizados alcanzó un valor de 0.752, considerando un total de 10 elementos evaluados. De acuerdo con los criterios de interpretación establecidos, estos valores se ubican dentro del rango de fiabilidad aceptable, lo que indica que el instrumento presenta un nivel adecuado de consistencia interna para evaluar la percepción de los docentes respecto al sistema. Por tanto, los resultados obtenidos a partir de la encuesta SUS pueden considerarse válidos y confiables para el análisis de la calidad en uso del aplicativo móvil.

3.3 Evaluación del modelo de calidad en uso

Una vez que se han tabulado los datos del taller práctico y encuesta SUS, como se observa en los Anexos C y D, se han calculado las diferentes métricas de la calidad en uso, definidas previamente en el modelo de la **Tabla 33**.

3.3.1 Eficacia

- Sub característica: Tareas completadas

En base a la siguiente formula:

$$X = \frac{A}{B}$$

Donde A es el número de tareas completadas y B el número de tareas. El resultado es el siguiente:

$$X = \frac{160}{160} = 1 = 100\%$$

Se alcanzó un 100% de efectividad, evidenciando que el sistema es completamente funcional y fácil de usar en situaciones reales.

- Sub característica: Objetivos completados

A diferencia de la Fase 1, donde cada objetivo comprendía múltiples tareas, en la Fase 2 se diseñó el taller práctico de modo que cada objetivo constara de una sola tarea, de tal manera que la realización del taller fuera más rápida e intuitiva para los docentes. Aplicando la misma fórmula de la sub característica anterior:

$$X = \frac{A}{B} = \frac{160}{160} = 1 = \mathbf{100\%}$$

Evidentemente, también se alcanzó un 100% de efectividad, demostrando que se cumplieron todos los objetivos exitosamente.

- Sub característica: Errores de una tarea

En base a la siguiente formula:

$$X = 1 - \frac{A}{B}$$

Donde A es el número de errores en tareas completadas y B es el número de tareas. Se obtuvo el siguiente resultado:

$$X = 1 - \frac{0}{160} = 1 = \mathbf{100\%}$$

Ningún docente identificó ningún error en las diferentes tareas, por lo que se alcanzó un 100% de tareas completadas sin errores.

3.3.2 Eficiencia

- Sub característica: Tiempo de tareas

En base a la siguiente formula:

$$X = \frac{A}{B}$$

Donde A es el tiempo empleado por usuarios expertos para completar una tarea, mientras que B es el tiempo empleado por los usuarios nuevos. Se calculó el tiempo promedio de los 5 usuarios expertos identificados. Posteriormente, se calculó el tiempo promedio de los 15 usuarios nuevos restantes. El resultado obtenido fue el siguiente:

$$X = \frac{10:18 \text{ min}}{13:15 \text{ min}} = 0.78 = \mathbf{78\%}$$

Al aplicar la fórmula, el resultado obtenido es de 0.78, lo que indica que los usuarios expertos tardan aproximadamente el 78% del tiempo que requieren los usuarios nuevos para completar las mismas tareas. En otras palabras, los usuarios nuevos tardan un 22% más de tiempo que los usuarios expertos.

- Sub característica: Eficiencia del tiempo

En base a la siguiente fórmula:

$$X = \frac{A}{B}$$

En donde A es el tiempo empleado por usuarios expertos para completar un objetivo, mientras que B es el tiempo empleado por los usuarios nuevos. Dado que en la Fase 2 se estableció una correspondencia de uno a uno entre objetivos y tareas (8 objetivos correspondientes a 8 tareas), a diferencia de la Fase 1 donde cada objetivo comprendía múltiples tareas, el tiempo promedio para completar un objetivo resulta equivalente al tiempo promedio para completar una tarea. En consecuencia, los valores obtenidos fueron idénticos a los de la sub característica de tiempo de tareas:

$$X = \frac{10:18 \text{ min}}{13:15 \text{ min}} = 0.78 = \mathbf{78\%}$$

Este resultado confirma que los usuarios expertos completan los objetivos del taller empleando aproximadamente el 77.8% del tiempo requerido por los usuarios nuevos, lo que evidencia una eficiencia aceptable del sistema y una curva de aprendizaje moderada para usuarios sin experiencia previa.

3.3.3 Satisfacción

- Sub característica: Utilidad

Para las respuestas de la encuesta SUS que usan la escala de Likert, se establecieron los mismos pesos que en la Fase 1, como se observa en la **Tabla 36**.

Tabla 36 Pesos de respuesta de encuesta SUS

Escala	Respuesta	Peso
1	Totalmente en desacuerdo	0.2
2	En desacuerdo	0.4
3	Neutro	0.6
4	De acuerdo	0.8
5	Totalmente de acuerdo	1

Las preguntas P1, P6 y P9 permiten calcular la utilidad. Cabe recalcar que fue necesario invertir la pregunta P6 para que todas tengan sentido positivo.

Tabla 37 Resultados de la encuesta para la utilidad

Pregunta	1	2	3	4	5	Resultado
P1	0	0	0	5	15	19 de 20
P6	0	1	0	4	15	18.6 de 20
P9	1	0	0	2	17	18.8 de 20
Promedio						18.8 de 20

En base a la siguiente fórmula:

$$X = \frac{A}{B}$$

Donde A se refiere a los usuarios que consideraron útil la aplicación móvil y B al total de docentes encuestados.

$$X = \frac{18.8}{20} = 0.94 = \mathbf{94\%}$$

Se obtuvo un 94% de utilidad, lo que implica que la gran mayoría de los docentes participantes (18.8 de 20 encuestados) considera que las nuevas funcionalidades de la Fase 2 son útiles para su gestión académica. Este resultado demuestra que el sistema cumple satisfactoriamente con su propósito de facilitar la gestión del portafolio docente, proporcionando una herramienta efectiva que responde a las necesidades reales de los usuarios y optimiza sus procesos académicos cotidianos.

- Sub característica: Confianza

En base a la siguiente fórmula:

$$X = 1 - \frac{A}{B}$$

Donde A es el número total de usuarios con reclamos y B el total de docentes encuestados. Se obtuvo el siguiente resultado:

$$X = 1 - \frac{8}{20} = 0.6 = \mathbf{60\%}$$

El 60% de confianza obtenido indica que 12 de los 20 docentes encuestados (60%) no realizaron observaciones críticas sobre el sistema, mientras que 8 docentes (40%) proporcionaron sugerencias de mejora. Este nivel de confianza se considera aceptable según la norma ISO/IEC 25022, reflejando que la mayoría de los usuarios confía en la aplicación y la encuentra adecuada para la gestión del portafolio docente. Las observaciones recibidas constituyen retroalimentación valiosa para futuras iteraciones del sistema.

- Sub característica: Comodidad

Las preguntas P2, P3, P4, P5, P7, P8 y P10 permiten calcular la comodidad. Al igual que se hizo en el cálculo de la utilidad, se invirtió el sentido de las preguntas P2, P4, P8 y P10 para que todas sean positivas.

Tabla 38 Resultados de la encuesta para la comodidad

Pregunta	1	2	3	4	5	Resultado
P2	0	3	4	3	10	16 de 20
P3	0	0	0	6	14	18.8 de 20
P4	0	3	0	4	13	17.4 de 20
P5	0	0	2	2	16	18.8 de 20
P7	0	0	0	6	14	18.8 de 20
P8	2	2	0	3	13	16.6 de 20
P10	0	2	1	2	15	18 de 20
Promedio						17.78 de 20

En base a la siguiente fórmula:

$$X = \frac{A}{B}$$

Donde A se refiere a los usuarios que consideraron que la aplicación móvil es cómoda y B al total de docentes encuestados.

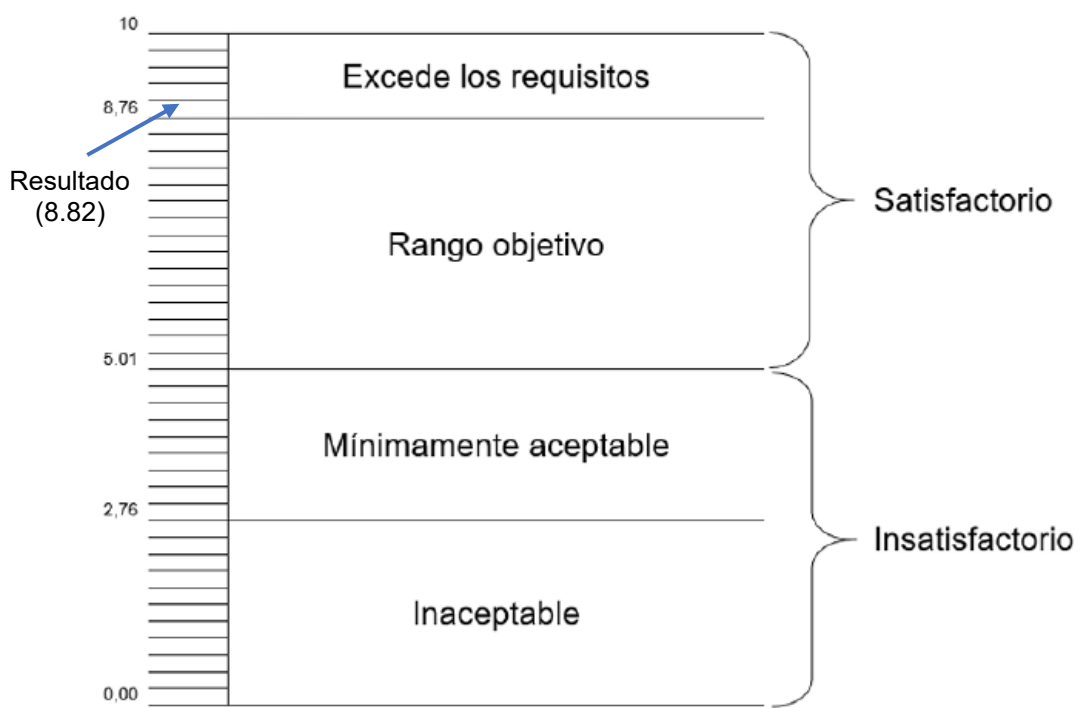
$$X = \frac{17.78}{20} = 0.89 = \mathbf{89\%}$$

Se obtuvo un 89% de comodidad, lo que demuestra un alto nivel de satisfacción con la usabilidad del sistema. Este porcentaje indica que la aplicación móvil presenta una interfaz intuitiva y funcionalidades bien integradas, permitiendo que la mayoría de los docentes (17.78 de 20 encuestados) la consideren fácil de usar sin necesidad de conocimientos técnicos avanzados. El resultado refleja que el diseño del sistema minimiza la complejidad y facilita el aprendizaje, contribuyendo a una experiencia de usuario positiva y eficiente.

3.4 Resultados del modelo de calidad en uso

Una vez calculadas las métricas de todas las subcaracterísticas definidas en la **Tabla 33**, se pudo validar con éxito la calidad en uso de la Fase 2 del portafolio docente. Como se aprecia en la **Tabla 39**, una vez aplicado el peso de característica a cada resultado, el resultado indica que se obtuvo un **88.28%** de la calidad en uso, ubicándose en el **rango satisfactorio** y además **excediendo los requisitos** según la norma ISO/IEC 25022, como lo muestra la **Figura 73**.

Figura 73 Rango de resultados de calidad en uso



Este resultado demuestra que la aplicación móvil cumple satisfactoriamente con los criterios establecidos de eficacia (42.00%), eficiencia (24.96%) y satisfacción (21.32%), confirmando que el sistema es efectivo, eficiente y aceptado por los docentes usuarios. El porcentaje obtenido valida la implementación técnica y funcional de la Fase 2 Portafolio Docente en UTN-Móvil como una solución de calidad para la gestión académica docente de la Universidad Técnica del Norte.

Tabla 39 Resultados de calidad en uso

Modelo de Calidad en Uso						
Característica	Sub Característica	Peso de Característica	Peso de Sub Característica	Medición	Resultado	Resultado de Característica
Eficacia	Tareas completadas	42%	16%	100%	16.00%	42.00%
	Objetivos completados		16%	100%	16.00%	
	Tareas sin errores		10%	100%	10.00%	
Eficiencia	Tiempo de tareas	32%	16%	78%	12.48%	24.96%
	Eficiencia de tiempo		16%	78%	12.48%	
Satisfacción	Utilidad	26%	10%	94%	9.40%	21.32%
	Confianza		8%	60%	4.80%	
	Comodidad		8%	89%	7.12%	
Total						88.28%

CONCLUSIONES

La definición del marco teórico permitió comprender en profundidad el contexto que engloba la Fase 2 del portafolio docente, así como las herramientas de desarrollo aplicadas, la metodología aplicada y el aporte de la norma ISO/IEC 25022 en la evaluación de la calidad en uso. Mediante la investigación en múltiples bases de datos científicas, se pudo consultar y contrastar información actualizada y de calidad.

La continuidad de la arquitectura de software definida en la Fase 1 resultó fundamental, debido a que permitió agilizar el trabajo de desarrollo. Además, dicha arquitectura demostró ser segura y escalable, al soportar la incorporación de nuevas funcionalidades robustas y el incremento de nuevos usuarios sin afectar notablemente el rendimiento y estabilidad del sistema.

El framework NestJS, utilizado para el backend, evidenció ser una herramienta fundamental para el desarrollo e implementación de microservicios, al ser robusta y escalable, además de tener una curva de aprendizaje accesible. Su enfoque basado en módulos e inyección de dependencias permitió agregar nuevas funcionalidades de manera intuitiva y segura. Asimismo, la arquitectura de tres capas establecida facilitó la clara separación de responsabilidades entre el *controller*, *service* y *repository*.

La utilización del framework Flutter para el frontend facilitó el desarrollo e implementación de una aplicación móvil multiplataforma eficiente y mantenible a lo largo del tiempo. Los principios de arquitectura limpia implementados mediante el establecimiento de las capas de dominio, infraestructura y presentación, facilitó la separación de responsabilidades y promovió la independencia entre la lógica de negocio y los componentes de interfaz.

La evaluación de la calidad en uso de la Fase 2, realizada a través de un taller práctico y la encuesta SUS, resultó valiosa, ya que permitió establecer contacto directo y recibir retroalimentación de múltiples docentes que probaron las nuevas funcionalidades de la aplicación móvil. La calidad en uso alcanzada en este trabajo de titulación fue del 88.28%, valor que, según la escala establecida, se ubica dentro del rango satisfactorio, e incluso excede los requisitos, lo que demuestra que la implementación de la nueva fase fue exitosa.

RECOMENDACIONES

Se recomienda seguir expandiendo y actualizando el servicio del portafolio docente en UTN-Móvil, ya sea en una posterior Fase 3 o mediante el trabajo del personal encargado del DDTI. Asimismo, es importante considerar la implementación de funcionalidades del portafolio docente del SIIU que sean realmente utilizadas por los docentes y estos vean conveniente su uso desde UTN-Móvil; entre ellas, aquellas que actualmente el servicio móvil carece, como la calificación de trabajos de titulación, proyectos de investigación y las relacionadas con vinculación.

En relación con la mejora del servicio del portafolio docente, tanto en el SIIU como en UTN-Móvil, resulta fundamental coordinar el trabajo de ambos equipos de desarrollo con el fin de mejorar funcionalidades actuales. Algunas mejoras pueden ser: agregar más tipos de pregunta, compartir actividades o evaluaciones mediante código QR, integrar el uso de inteligencia artificial para detectar plagio en tareas o para generar resúmenes de proyectos de investigación u obras de producción científica. Estas y más posibilidades permitirían ofrecer un servicio de aún mejor calidad a la comunidad universitaria.

Considerando que actualmente todos los microservicios de UTN-Móvil, con excepción del API Gateway, corren en el mismo servidor, resulta evidente que la solución vigente puede presentar problemas graves de disponibilidad y estabilidad, especialmente en los servicios más utilizados, como el portafolio estudiante y docente. Este enfoque puede optimizarse mediante la separación de los microservicios más utilizados en servidores dedicados. Adicionalmente, se pueden aplicar técnicas de escalado horizontal, como balanceo de carga o replicación de microservicios, o simplemente escalado vertical, agregando más recursos computacionales al servidor actual.

Para futuros trabajos de desarrollo de software de UTN-Móvil, se recomienda profundizar en la aplicación de la norma ISO/IEC 25022 para la evaluación de la calidad en uso. Principalmente mediante la evaluación de las métricas faltantes. Estas son la libertad de riesgo, evaluando aspectos críticos como el riesgo económico o de salud, y la cobertura del contexto, midiendo factores como la flexibilidad del sistema a distintos dispositivos móviles con diversas características.

REFERENCIAS

- [1] I. N. de E. y Censos, «Indicadores de TIC 2022 – 2024», Instituto Nacional de Estadística y Censos, 2024. [En línea]. Disponible en: <https://www.ecuadorencifras.gob.ec/tecnologias-de-la-informacion-y-comunicacion-tic/>
- [2] D. M. D. Oliveira, L. Pedro, y C. Santos, «The use of mobile applications in higher education classes: a comparative pilot study of the students' perceptions and real usage», *Smart Learn. Environ.*, vol. 8, n.º 1, p. 14, 2021, doi: 10.1186/s40561-021-00159-6.
- [3] F. L. Enríquez Rosero, «Implementación del servicio “Portafolio Docentes” en la plataforma móvil de la Universidad Técnica del Norte a través del marco de trabajo Scrum y la Norma ISO/IEC 25022», Universidad Técnica del Norte, 2025. Accedido: 4 de junio de 2025. [En línea]. Disponible en: <https://repositorio.utn.edu.ec/handle/123456789/17222>
- [4] A. C. Sassa, I. A. de Almeida, T. N. F. Pereira, y M. S. de Oliveira, «Scrum: A Systematic Literature Review», *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, n.º 4, 2023, doi: 10.14569/IJACSA.2023.0140420.
- [5] R. A. Huertas Burgos, «Desarrollo de una guía metodológica para mejorar la documentación del capítulo técnico de los trabajos de grado de la carrera de Software que utilicen la metodología SCRUM», bachelorThesis, 2024. Accedido: 5 de junio de 2025. [En línea]. Disponible en: <https://repositorio.utn.edu.ec/handle/123456789/15812>
- [6] B. Jaradat y G. S. Weheba, «A practical guide to ISO 25022: Measurement of software quality in-use», *J. Manag. Eng. Integr.*, vol. 17, n.º 1, pp. 47-54, 2024, doi: 10.62704/10057/28085.
- [7] O. de las N. Unidas, «Objetivo 9: Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación». 2025. [En línea]. Disponible en: <https://www.un.org/sustainabledevelopment/es/infrastructure/>
- [8] J. Suthendra y M. Pakereng, «Implementation of Microservices Architecture on E-Commerce Web Service», *ComTech Comput. Math. Eng. Appl.*, vol. 11, pp. 89-95, dic. 2020, doi: 10.21512/comtech.v11i2.6453.
- [9] A. D. Pham, «Developing back-end of a web application with NestJS framework: Case: Integrify Oy's student management system», PhD Thesis, 2020. [En línea]. Disponible en: <https://www.theseus.fi/handle/10024/353200>
- [10] S. A. Kinari, N. Funabiki, S. T. Aung, K. H. Wai, M. Mentari, y P. Puspitaningayu, «An Independent Learning System for Flutter Cross-Platform Mobile Programming with Code Modification Problems», *Information*, vol. 15, n.º 10, 2024, doi: 10.3390/info15100614.
- [11] C. Buabeng-Andoh, «Investigating student-teachers' continuous intention to use mobile learning management system: the technology acceptance model and expectation confirmatory model», *Discov. Educ.*, vol. 4, n.º 1, p. 76, 2025, doi: 10.1007/s44217-025-00447-0.
- [12] S. M. Fierro Dávila, «Implementación del servicio “Portafolio Estudiante” en la plataforma móvil de la universidad técnica del norte utilizando el marco de trabajo SCRUM y la norma ISO/IEC 25022», bachelorThesis, 2025. Accedido: 25 de junio de 2025. [En línea]. Disponible en: <https://repositorio.utn.edu.ec/handle/123456789/16969>
- [13] I. A. L. Vásquez, «Desarrollo de una aplicación móvil utilizando el framework Flutter para fomentar el área turística del GAD de Pedro Moncayo», PhD Thesis, Universidad Técnica del Norte, 2022. [En línea]. Disponible en: <https://repositorio.utn.edu.ec/handle/123456789/13228>
- [14] J. Jordanov, D. Simeonidis, y P. Petrov, «Containerized Microservices for Mobile Applications Deployed on Cloud Systems», *Int. J. Interact. Mob. Technol. IJIM*, vol. 18, n.º 10, pp. 48-58, may 2024, doi: 10.3991/ijim.v18i10.45929.

- [15] C. Guevara-Vega, J. Hernández-Rojas, M. Botto-Tobar, I. García-Santillán, A. Basantes-Andrade, y A. Quiña-Mera, «Automation of the Municipal Inspection Process in Ecuador Applying Mobile-D for Android», 2020, pp. 155-166. doi: 10.1007/978-3-030-32022-5_15.
- [16] A. Laswi, A. Apriyanto, y A. A.M., «Implementation of Flutter-based Learning Management System (LMS) at Universitas Andi Djemma Palopo», *PENA Tek. J. Ilm. Ilmu-Ilmu Tek.*, vol. 7, p. 1, mar. 2022, doi: 10.51557/pt_jiit.v7i1.1173.
- [17] M. F. García Guerra, «Automatización del subproceso para aprobación de proyectos de investigación, integrada al SIU de la UTN», bachelorThesis, 2024. Accedido: 5 de junio de 2025. [En línea]. Disponible en: <https://repositorio.utn.edu.ec/handle/123456789/15697>
- [18] Universidad Técnica del Norte, *Reglamento de Portafolio Universitario*. 2014, p. 11. [En línea]. Disponible en: https://legislacion.utn.edu.ec/wp-content/uploads/2017/12/2.REGLAMENTO_PORTAFOLIO_UNIVERSITARIO-1.pdf
- [19] «Oracle APEX», Oracle APEX. Accedido: 5 de junio de 2025. [En línea]. Disponible en: <https://apex.oracle.com/en/>
- [20] «Why Oracle APEX», Oracle APEX. Accedido: 9 de julio de 2025. [En línea]. Disponible en: <https://apex.oracle.com/en/platform/why-oracle-apex/>
- [21] I. Pastierik, «Deploying Oracle Machine Learning AutoML Models for Oracle APEX Analytics», en *2024 IEEE 17th International Scientific Conference on Informatics (Informatics)*, nov. 2024, pp. 499-506. doi: 10.1109/Informatics62280.2024.10900807.
- [22] «Database Concepts». Accedido: 5 de junio de 2025. [En línea]. Disponible en: https://docs.oracle.com/cd/E11882_01/server.112/e40540/intro.htm#CNCPT958
- [23] X. Liao *et al.*, «Performance Diagnosis of Oracle Database Systems Based on Image Encoding and VGG16 Model», *IEEE Access*, vol. 12, pp. 137194-137202, 2024, doi: 10.1109/ACCESS.2024.3448261.
- [24] T. I. Mohottige, A. Polyvyanny, C. Fidge, R. Buyya, y A. Barros, «Reengineering software systems into microservices: State-of-the-art and future directions», *Inf. Softw. Technol.*, vol. 183, p. 107732, jul. 2025, doi: 10.1016/j.infsof.2025.107732.
- [25] D. Wang, H. Shen, H. Tian, y Y. Yang, «Schedule multi-instance microservices to minimize response time under budget constraint in cloud HPC systems», *J. Parallel Distrib. Comput.*, vol. 202, p. 105086, ago. 2025, doi: 10.1016/j.jpdc.2025.105086.
- [26] Amazon, «What are Microservices? | AWS», Amazon Web Services, Inc. Accedido: 4 de junio de 2025. [En línea]. Disponible en: <https://aws.amazon.com/microservices/>
- [27] P. Antunes y N. Guimarães, «Guiding the implementation of data privacy with microservices», *Int. J. Inf. Secur.*, vol. 23, n.º 6, pp. 3591-3608, dic. 2024, doi: 10.1007/s10207-024-00907-y.
- [28] «Architectural Styles and the Design of Network-based Software Architectures». Accedido: 5 de junio de 2025. [En línea]. Disponible en: <https://ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [29] A. Lercher, «Managing API Evolution in Microservice Architecture», en *2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, abr. 2024, pp. 195-197. doi: 10.1145/3639478.3639800.
- [30] «Technology | 2024 Stack Overflow Developer Survey». Accedido: 8 de junio de 2025. [En línea]. Disponible en: <https://survey.stackoverflow.co/2024/technology>
- [31] Microsoft, «Handbook - The TypeScript Handbook». Accedido: 19 de julio de 2025. [En línea]. Disponible en: <https://www.typescriptlang.org/docs/handbook/intro.html>
- [32] «Node.js — Introduction to TypeScript». Accedido: 19 de julio de 2025. [En línea]. Disponible en: <https://nodejs.org/en/learn/typescript/introduction>
- [33] G. M. Bierman, M. Abadi, y M. Torgersen, «Understanding TypeScript», en *European Conference on Object-Oriented Programming*, 2014. [En línea]. Disponible en: <https://api.semanticscholar.org/CorpusID:45852881>

- [34] «NestJS - A progressive Node.js framework», Documentation | NestJS - A progressive Node.js framework. Accedido: 4 de junio de 2025. [En línea]. Disponible en: <https://docs.nestjs.com>
- [35] M. Cosgaya Capel, «Creation of a middleware for blockchain interaction based on NestJS», *Univ. Politècnica Catalunya*, jul. 2023, Accedido: 18 de junio de 2025. [En línea]. Disponible en: <https://hdl.handle.net/2117/402646>
- [36] «Modules | Documentation | NestJS - A progressive Node.js framework», Documentation | NestJS - A progressive Node.js framework. Accedido: 10 de junio de 2025. [En línea]. Disponible en: <https://docs.nestjs.com/modules>
- [37] «Providers | Documentation | NestJS - A progressive Node.js framework», Documentation | NestJS - A progressive Node.js framework. Accedido: 10 de junio de 2025. [En línea]. Disponible en: <https://docs.nestjs.com/providers>
- [38] «Pipes | Documentation | NestJS - A progressive Node.js framework», Documentation | NestJS - A progressive Node.js framework. Accedido: 7 de febrero de 2026. [En línea]. Disponible en: <https://docs.nestjs.com/pipes>
- [39] «Controllers | Documentation | NestJS - A progressive Node.js framework», Documentation | NestJS - A progressive Node.js framework. Accedido: 14 de junio de 2025. [En línea]. Disponible en: <https://docs.nestjs.com/controllers>
- [40] A. Alhaddad, A. Andrews, y Z. Abdalla, «Chapter One - FSMApp: Testing mobile apps», en *Advances in Computers*, vol. 130, A. R. Hurson, Ed., Elsevier, 2023, pp. 1-65. doi: 10.1016/bs.adcom.2022.09.001.
- [41] R. Islam, R. Islam, y T. Mazumder, «Mobile application and its global impact», *Int. J. Eng. Technol.*, vol. 10, n.º 6, 2010.
- [42] M. Nagappan y E. Shihab, «Future Trends in Software Engineering Research for Mobile Apps», en *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 2016, pp. 21-32. doi: 10.1109/SANER.2016.88.
- [43] M. F. Ramlan y M. K. M. Nasir, «The Impact of Mobile Applications in Education: A Concept Paper», *Int. J. Acad. Res. Progress. Educ. Dev.*, vol. 12, n.º 3, pp. 897-904, sep. 2023, doi: 10.6007/IJARPED/v12-i3/19609.
- [44] E. A. Marín Lozano y D. A. López Lunagómez, «Aplicaciones Móviles para Mejorar la Comunicación entre Docente y Alumno», *Interconectando Saberes*, vol. 0, n.º 8, nov. 2019, doi: 10.25009/is.v0i8.2645.
- [45] A. Patidar y U. Suman, «Towards Analyzing Mobile App Characteristics for Mobile Software Development», en *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2021, pp. 786-790.
- [46] A. Ahmad, K. Li, C. Feng, S. M. Asim, A. Yousif, y S. Ge, «An Empirical Study of Investigating Mobile Applications Development Challenges», *IEEE Access*, vol. 6, pp. 17711-17728, 2018, doi: 10.1109/ACCESS.2018.2818724.
- [47] IBM, «Mobile app types», Mobile app types - IBM Documentation. Accedido: 19 de julio de 2025. [En línea]. Disponible en: <https://www.ibm.com/docs/en/maas360?topic=apps-mobile-app-types>
- [48] A. Ebone, Y. Tan, y X. Jia, «A Performance Evaluation of Cross-Platform Mobile Application Development Approaches», en *2018 IEEE/ACM 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, jun. 2018, pp. 92-93.
- [49] «Flutter | Multi-Platform». Accedido: 4 de junio de 2025. [En línea]. Disponible en: <https://flutter.dev/multi-platform/>
- [50] «Flutter architectural overview». Accedido: 6 de julio de 2025. [En línea]. Disponible en: <https://docs.flutter.dev/resources/architectural-overview>
- [51] A. Nirumand Jazi, I. Alfonso, y J. Cabot, «Low-Code Flutter Application Development Solution», en *Proceedings of the ACM/IEEE 27th International Conference on Model Driven*

- Engineering Languages and Systems*, en *MODELS Companion '24*. New York, NY, USA: Association for Computing Machinery, 2024, pp. 838-847. doi: 10.1145/3652620.3688330.
- [52] «Widgets». Accedido: 6 de julio de 2025. [En línea]. Disponible en: <https://docs.flutter.dev/get-started/fundamentals/widgets>
- [53] «Dart overview». Accedido: 5 de junio de 2025. [En línea]. Disponible en: <https://dart.dev/overview/>
- [54] I. David y R. Gelbard, «Using sentiment analysis to assess PMBOK knowledge areas' compatibility with agile methodology», *CENTERIS – Int. Conf. Enterp. Inf. Syst. ProjMAN - Int. Conf. Proj. Manag. HCist - Int. Conf. Health Soc. Care Inf. Syst. Technol.* 2023, vol. 239, pp. 381-395, ene. 2024, doi: 10.1016/j.procs.2024.06.185.
- [55] «Manifiesto for Agile Software Development». Accedido: 5 de junio de 2025. [En línea]. Disponible en: <https://agilemanifesto.org/>
- [56] G. Neri, R. Marchand, y N. Walkinshaw, «Exploratory Software Testing in Scrum: A Qualitative Study», en *Agile Processes in Software Engineering and Extreme Programming*, S. Peter, M. Kropp, A. Aguiar, C. Anslow, M. I. Lunesu, y A. Pinna, Eds., Cham: Springer Nature Switzerland, 2025, pp. 160-175.
- [57] J. Sutherland, *Jeff Sutherland's Scrum Handbook*. 2010.
- [58] K. Schwaber y J. Sutherland, *The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game*. 2020. [En línea]. Disponible en: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- [59] A. Cockburn, *Agile Software Development*. 2002, p. 201.
- [60] V. Nguyen, «Scrum Methodology: A proven formula for managing complex projects», Rabiloo. Accedido: 4 de febrero de 2026. [En línea]. Disponible en: <https://rabiloo.com/blog/scrum-methodology-a-proven-formula-for-managing-complex-projects>
- [61] «ISO/IEC 25010:2023», ISO. Accedido: 5 de junio de 2025. [En línea]. Disponible en: <https://www.iso.org/standard/78176.html>
- [62] A. Abran, W. Suryn, A. Khelifi, J. Rilling, A. Seffah, y F. Robert, «Consolidating the ISO Usability Models», ene. 2004.
- [63] «La familia de normas ISO/IEC 25000», ISO 25000. Accedido: 20 de julio de 2025. [En línea]. Disponible en: <https://iso25000.com/index.php/normas-iso-25000>
- [64] «ISO/IEC 25022:2016», ISO. Accedido: 5 de junio de 2025. [En línea]. Disponible en: <https://www.iso.org/standard/35746.html>
- [65] P. F. Oliveira y P. Matos, «Mobile Integration for Sakai LMS and Class Schedule Platform at an Higher Education Institution», en *2023 International Conference on Engineering and Emerging Technologies (ICEET)*, oct. 2023, pp. 1-6. doi: 10.1109/ICEET60227.2023.10525775.
- [66] B. A. Aldás Acosta, «Aplicación móvil educativa usando el Framework Flutter para el aprendizaje de lengua de señas ecuatoriana en la ciudad de Ambato.», sep. 2023, Accedido: 17 de junio de 2025. [En línea]. Disponible en: <https://repositorio.uta.edu.ec/handle/123456789/39444>
- [67] Q. Li y L. Zhang, «Design and Implementation of Online Education Platform with Microservice Architecture», en *2024 IEEE 15th International Conference on Software Engineering and Service Science (ICSESS)*, sep. 2024, pp. 173-177. doi: 10.1109/ICSESS62520.2024.10719050.
- [68] B. D. Morales Toapanta, «Construcción de una aplicación móvil para evaluar el desempeño docente de la Universidad Politécnica Salesiana», bachelorThesis, 2024. Accedido: 18 de junio de 2025. [En línea]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/26782>
- [69] L. H. Aung, S. T. Aung, N. Funabiki, H. H. S. Kyaw, y W.-C. Kao, «An Implementation of Web-Based Answer Platform in the Flutter Programming Learning Assistant System Using Docker Compose», *Electronics*, vol. 13, n.º 24, 2024, doi: 10.3390/electronics13244878.

- [70] J. Sulla-Torres, A. Gutierrez-Quintanilla, H. Pinto-Rodriguez, R. Gómez-Campos, y M. Cossio-Bolaños, «Quality in use of an android-based mobile application for calculation of bone mineral density with the standard ISO/IEC 25022», *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, n.º 8, pp. 158-163, 2020, doi: 10.14569/IJACSA.2020.0110821.
- [71] A. R. Ahmed, M. Tayyab, D. S. N. Bhatti, D. A. J. Alzahrani, y D. M. I. Babar, «Impact of Story Point Estimation on Product using Metrics in Scrum Development Process», *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, n.º 4, 2017, doi: 10.14569/IJACSA.2017.080452.

ANEXOS

Anexo A. Fotografías de la evaluación a docentes



Anexo B. Taller práctico para evaluar la Fase 2

FACULTAD	PICA
NOMBRE	

INDICACIONES:

1	Registre en la tabla con la hora de inicio y la hora de fin de cada tarea (Ctrl + Shift + T)
2	Indique si la aplicación mostró algún error durante la tarea (Si/No)
3	Indique si completó la tarea con éxito (Si/No)
Nota:	Inicie cada objetivo desde la pantalla principal del Portafolio Docente

NOTA: Todas las tareas se realizarán en el Portafolio Docente (Abrir UTN-Móvil -> Ingresar al Portafolio Docente)

Objetivo	Nro.	Descripción de la tarea	Inicio	Fin	Total	Errores	Finalizó	Observación
Ver pagos	1	Desde el menú principal, seleccionar la opción Más y luego Pagos: - Seleccionar un pago para expandirlo y ver sus conceptos. - Seleccionar Factura para visualizar el documento PDF.			0:00:00			
Ver control de bienes	2	Seleccionar el menú Más y luego Control de Bienes: - Seleccionar Reporte. - Seleccionar Reporte de activos.			0:00:00			
Ver Proyecto de Investigación	3	Seleccionar el menú Más y luego Proyectos de Investigación: - Seleccionar 'Mis Proyectos' - Seleccionar el primer proyecto del listado - Escoger la pestaña de Equipo y visualizar el listado de miembros.			0:00:00			
Ver Obra de Producción Científica	4	Ingresar al menú Más y luego Producción Científica: - Seleccionar Artículos (En caso de no tener escoger otro tipo de obra) - Seleccionar primer obra del listado - Visualizar cualquiera de las pestañas del detalle de obra			0:00:00			
Comentar en Foro General	5	Ingresar al Registro Académico, luego al Aula Virtual de una materia: - Seleccionar Foro General (la última pestaña de la barra superior). - Seleccionar botón de crear (esquina inferior derecha). - Escribir comentario - Enviar comentario			0:00:00			
Crear Evaluación Online	6	Dentro del Aula Virtual de la materia, ir a la pestaña Actividades (primera pestaña de la barra superior): - Seleccionar el botón Crear - Llenar título, instrucciones y contenido - Seleccionar Primera Parcial - En política de evaluación escoger Pruebas (30%) - En tipo de actividad escoger Evaluación Online (Dentro de Plataforma Institucional) - En configuración de evaluación online escoger Manual - Establecer límite de tiempo de evaluación - Crear actividad			0:00:00			
Crear 3 preguntas	7	Desde la pestaña Actividades del Aula virtual, seleccionar la nueva actividad creada en la sección de Pruebas: - Escoger la opción Agregar Preguntas. - Seleccionar el botón Crear. - Escoger como tipo de pregunta Respuesta Única. - Seleccionar cualquier contenido y nivel de dificultad. - Escribir enunciado. - Escribir alternativas de respuesta, marcar al menos una como correcta. - Guardar pregunta. - Repetir proceso y crear 2 preguntas más.			0:00:00			
Agregar Preguntas a Evaluación Online	8	Dentro de la página Agregar Preguntas, expandir la sección de Preguntas Disponibles: - Seleccionar las tres preguntas creadas recientemente - Seleccionar acción de Agregar (debajo del título Preguntas Disponibles) - Confirmar acción de agregar 3 preguntas - Visualizar preguntas agregadas en sección Preguntas Agregadas			0:00:00			

Final	Una vez finalizado el taller guárdelo y complete la encuesta. https://forms.office.com/r/7MpiTWy8kz2orjein-lorLink
-------	--

Total 0:00:00

Anexo C. Resultados del Taller Práctico

Id	Facultad	T1	T2	T3	T4	T5	T6	T7	T8	Errores	Tareas Logradas	Tiempo Total	Reclamos
2	FCCSS	01:20	00:16	00:51	01:03	00:53	02:39	07:03	01:38	0	8	15:43	NO
3	FCCSS	01:03	00:42	01:06	00:16	00:29	02:30	08:58	00:37	0	8	15:41	NO
4	FICAYA	01:26	00:48	01:05	00:49	02:12	02:23	05:52	02:22	0	8	16:57	NO
5	FCCSS	01:00	01:00	02:00	01:00	01:00	02:00	03:20	00:40	0	8	12:00	NO
6	FCCSS	00:42	01:04	01:01	01:03	01:25	02:59	03:40	02:42	0	8	14:36	NO
7	FECYT	01:14	01:28	01:15	01:41	01:44	01:30	03:22	01:22	0	8	13:36	NO
8	FICAYA	00:30	00:35	00:40	00:30	01:00	05:00	06:00	01:00	0	8	15:15	SI
9	FECYT	01:00	01:00	01:20	01:35	01:12	02:51	04:54	02:44	0	8	16:36	NO
10	FICAYA	00:30	00:30	00:30	00:30	00:40	02:00	05:00	06:00	0	8	15:40	SI
11	FECYT	01:12	01:01	01:16	01:35	02:35	02:28	04:16	01:14	0	8	15:37	NO
12	FICA	01:45	00:33	00:45	00:52	00:27	01:46	03:39	00:57	0	8	10:44	NO
13	FICA	00:45	00:37	00:40	00:26	00:39	01:38	04:52	00:44	0	8	10:21	SI
14	FACAE	00:48	00:46	00:34	00:24	00:45	02:15	03:16	00:45	0	8	09:33	NO
15	FACAE	00:04	00:12	01:08	00:46	01:05	00:50	03:13	01:26	0	8	08:44	NO
16	FACAE	00:33	00:30	00:35	00:30	00:35	00:40	06:10	00:10	0	8	09:43	NO
17	FICA	00:38	00:35	01:30	00:45	00:50	02:00	02:55	01:25	0	8	10:38	NO
18	FICA	01:00	00:00	01:00	01:00	00:00	01:00	03:00	01:00	0	8	08:00	NO
19	FACAE	00:51	00:19	00:29	00:46	00:21	01:36	02:12	02:10	0	8	08:44	NO
20	FACAE	00:46	00:41	00:48	01:02	01:48	02:21	05:13	02:42	0	8	15:21	NO
21	FECYT	00:42	00:25	00:42	00:20	00:37	01:57	01:32	00:29	0	8	06:44	NO

Anexo D. Resultados de la encuesta SUS

ID	Facultad	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Observaciones
2	FCCSS	5	1	5	1	5	1	5	1	5	1	NO
3	FCCSS	5	1	5	2	5	1	5	1	5	1	SI
4	FICAYA	5	1	5	1	5	1	5	5	5	1	NO
5	FCCSS	4	1	5	1	4	1	4	1	1	1	SI
6	FCCSS	5	3	5	1	5	1	5	1	5	3	SI
7	FECYT	5	4	4	4	5	4	5	5	5	4	NO
8	FICAYA	5	1	5	4	5	1	5	1	5	2	NO
9	FECYT	5	4	4	1	5	2	4	4	5	1	NO
10	FICAYA	4	3	4	1	3	2	4	2	4	1	SI
11	FECYT	5	1	5	1	5	1	5	1	5	1	NO
12	FICA	4	3	5	2	3	2	4	2	4	2	NO
13	FICA	4	1	4	1	5	1	5	1	5	1	NO
14	FACAE	5	4	4	4	5	1	5	4	5	1	SI
15	FACAE	5	3	5	2	5	1	5	1	5	1	SI
16	FACAE	5	1	5	1	5	1	5	1	5	1	NO
17	FICA	4	2	5	1	4	1	5	1	5	1	NO
18	FICA	5	2	5	1	5	1	4	1	5	1	SI
19	FACAE	5	1	5	1	5	1	5	1	5	1	NO
20	FACAE	5	2	4	2	5	2	4	2	5	4	NO
21	FECYT	5	1	5	1	5	1	5	1	5	1	SI