

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS CARRERA DE SOFTWARE

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

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

AUTOR:

Santiago Marcelo Fierro Dávila

DIRECTOR:

Ing. José Antonio Quiña Mera, Ph.D.

Ibarra, 2025



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

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

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información.

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1003668934		
APELLIDOS Y NOMBRES:	FIERRO DÁVILA SANTIAGO MARCELO		
DIRECCIÓN:	IBARRA, AV. TEODORO GÓMEZ DE LA TORRE		
EMAIL:	smfierrod@utn.edu.ec / santiagofierrodavila@gmail.com		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0997252583

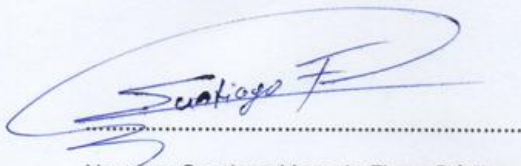
DATOS DE LA OBRA	
TÍTULO:	IMPLEMENTACIÓN DEL SERVICIO "PORTAFOLIO ESTUDIANTE" EN LA PLATAFORMA MÓVIL DE LA UNIVERSIDAD TÉCNICA DEL NORTE UTILIZANDO EL MARCO DE TRABAJO SCRUM Y LA NORMA ISO/IEC 25022.
AUTOR (ES):	FIERRO DÁVILA SANTIAGO MARCELO
FECHA DE APROBACIÓN: DD/MM/AAAA	24/02/2025
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO 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 24 días del mes de febrero de 2025

EL AUTOR:



Nombre: Santiago Marcelo Fierro Dávila

CI: 1003668934



UNIVERSIDAD TÉCNICA DEL NORTE BIBLIOTECA UNIVERSITARIA

CERTIFICACIÓN DEL DIRECTOR

Certifico que el trabajo de grado "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", ha sido desarrollado en su totalidad por el señor: Santiago Marcelo Fierro Dávila, portador de la cédula de identidad número 1003668934.

**JOSE
ANTONIO
QUIÑA MERA**



Firmado digitalmente por JOSE
ANTONIO QUIÑA MERA
Nombre de reconocimiento (DN): c=EC,
sn=QUIÑA MERA, givenName=JOSE
ANTONIO,
serialNumber=DCEC-1002322384,
cn=JOSE ANTONIO QUIÑA MERA,
2.5.4.97=TINEC-1002322384001

Ing. Quiña Mera José Antonio, Ph.D.

Director de Trabajo de Grado

Dedicatoria

A mis pilares, aquellos que con amor, paciencia y sacrificio han estado a mi lado en cada paso de este camino.

A mi madre, Rocío, por su inagotable amor, su fortaleza y por enseñarme que los sueños se alcanzan con esfuerzo y perseverancia. A mi padre, Renán, por ser un ejemplo de constancia y dedicación, y por brindarme siempre su apoyo incondicional. A mi hermana, Vitiam, por ser mi compañera de vida, mi cómplice y mi motivación.

A toda mi familia, mis tías, mi abuela, mis primos y primas, por su cariño y por ser parte de mi historia, por celebrar mis logros y darme fuerzas cuando más lo he necesitado.

A mis amigos, aquellos de siempre y los que la vida universitaria me regaló, por las risas, el apoyo y los momentos compartidos que hicieron de este camino una experiencia inolvidable.

A todos ustedes, con gratitud y amor, les dedico este logro.

Agradecimiento

Al llegar al final de este camino, no puedo sino sentir una profunda gratitud hacia todas las personas que han sido parte de este proceso y que, de una u otra forma, han dejado su huella en mi formación.

En primer lugar, a mis padres, Renán y Rocío, por su amor incondicional, por ser mi mayor fuente de inspiración y apoyo, y por enseñarme que con esfuerzo y perseverancia los sueños pueden hacerse realidad. A mi hermana, Vitiam, por su compañía y su comprensión.

A mi tutor de tesis, Antonio, por su guía y su disposición para orientarme en este desafío académico. A mi asesora, Cathy, por sus valiosos consejos y por ayudarme a dar forma a mis ideas con claridad y precisión.

A todos mis profesores universitarios, quienes, con su dedicación y conocimiento, han contribuido significativamente a mi formación profesional, inculcándome el amor por el aprendizaje y el deseo de seguir creciendo.

A cada uno de ustedes, mi más sincero agradecimiento.

Tabla de Contenidos

INTRODUCCIÓN	1
PLANTEAMIENTO DEL PROBLEMA.....	1
OBJETIVOS.....	2
<i>Objetivo General</i>	2
<i>Objetivos Específicos</i>	2
ALCANCE	2
METODOLOGÍA	4
JUSTIFICACIÓN.....	5
<i>Justificación Institucional y Tecnológica</i>	5
<i>Justificación Social</i>	5
CONTEXTO DE LA INVESTIGACIÓN.....	5
I MARCO TEÓRICO	8
1.1 SISTEMA INFORMÁTICO INTEGRADO UNIVERSITARIO DE LA UNIVERSIDAD TÉCNICA DEL NORTE	
8	
1.1.1 <i>Portafolio Estudiante UTN</i>	8
1.1.2 <i>Tecnologías utilizadas para el desarrollo</i>	8
1.1.3 <i>Otras plataformas universitarias similares en Ecuador</i>	9
1.2 APLICACIONES MÓVILES	10
1.2.1 <i>Tendencia de los usuarios hacia el uso de plataformas móviles</i>	10
1.2.2 <i>Aplicaciones en entornos Universitarios</i>	11
1.2.3 <i>Aplicaciones móviles multiplataforma</i>	11
1.2.4 <i>Lenguaje de programación Dart</i>	12
1.2.5 <i>Desarrollo de aplicaciones móviles multiplataforma con Flutter</i>	12
1.2.6 <i>Arquitectura limpia en Flutter</i>	12
1.3 ARQUITECTURAS MICROSERVICIOS	15
1.3.1 <i>Interfaz de programación de aplicaciones (API)</i>	15
1.3.2 <i>Plataformas microservicios</i>	15
1.3.1 <i>Selección del Lenguaje de programación</i>	15
1.3.2 <i>Lenguaje de programación TypeScript</i>	17
1.3.3 <i>Desarrollo de aplicación microservicios con NestJS</i>	17
1.4 METODOLOGÍA SCRUM	18
1.4.1 <i>Fase 1. Pre-Juego</i>	19
1.4.2 <i>Fase 2. Juego</i>	20
1.4.3 <i>Fase 3. Post-Juego</i>	22
1.5 MODELO DE EVALUACIÓN DE CALIDAD.....	22

1.5.1	<i>Norma ISO/IEC 25010</i>	23
1.5.2	<i>Calidad en Uso</i>	23
1.5.3	<i>Norma ISO/IEC 25022</i>	23
1.5.4	<i>Escala de medición de Calidad en Uso</i>	24
1.6	TRABAJOS RELACIONADOS.....	25
1.7	RESULTADOS OBTENIDOS DEL ESTUDIO.....	27
II	DESARROLLO	29
2.1	DESARROLLO FASE DE PRE-JUEGO	29
2.1.1	<i>Equipo SCRUM</i>	29
2.1.2	<i>Elicitación de Requerimientos</i>	30
2.2	DESARROLLO FASE DE JUEGO	37
2.2.1	<i>Sprint 0</i>	37
2.2.2	<i>Sprint 1</i>	45
2.2.3	<i>Sprint 2</i>	52
2.2.4	<i>Sprint 3</i>	57
2.2.5	<i>Sprint 4</i>	63
2.3	DESARROLLO FASE DE POST-JUEGO.....	68
III	VALIDACIÓN DE RESULTADOS	69
3.1	DEFINICIÓN DE MODELO DE CALIDAD EN USO	69
3.2	MEDICIÓN DE CALIDAD EN USO	70
3.2.1	<i>Definición de Muestra poblacional</i>	70
3.2.2	<i>Taller Práctico</i>	70
3.2.3	<i>Encuesta SUS</i>	72
3.3	VALIDACIONES ESTADÍSTICAS DE LOS INSTRUMENTOS DE MEDICIÓN	73
3.3.1	<i>Fiabilidad del taller práctico</i>	73
3.3.2	<i>Fiabilidad de la encuesta SUS</i>	74
3.4	EVALUACIÓN DEL MODELO DE CALIDAD EN USO.....	74
3.4.1	<i>Característica: Eficacia</i>	74
1.1.2	<i>Característica: Eficiencia</i>	76
1.1.3	<i>Característica: Satisfacción</i>	77
3.5	RESULTADOS DEL MODELO DE CALIDAD EN USO	80
	CONCLUSIONES	82
	RECOMENDACIONES	83
	REFERENCIAS	84
	ANEXOS	94

ANEXO A. FOTOGRAFÍAS DE APLICACIÓN DEL TALLER PRÁCTICO Y ENCUESTA SUS	94
ANEXO B. TALLER PRÁCTICO PARA EVALUAR LA APLICACIÓN MÓVIL	95
ANEXO C. TABULACIÓN DEL TALLER PRÁCTICO	96
ANEXO D. TABULACIÓN ENCUESTA SUS.....	103

Índice de Figuras

Fig. 1 Árbol de Problemas	1
Fig. 2 Arquitectura Propuesta.....	3
Fig. 3 Diagrama de Arquitectura Limpia [36]	13
Fig. 4 Resultados compración tecnologías	16
Fig. 5. Proceso de Scrum [61]	21
Fig. 6 División de la norma ISO/IEC 25000 [63].....	22
Fig. 7 Escala de medición de Calidad en Uso [69]	25
Fig. 8 Diagrama Fases de Desarrollo.....	29
Fig. 9 Arquitectura Plataforma Microservicios.....	38
Fig. 10 Estructura Arquitectónica API Microservicio	40
Fig. 11 Estructura Arquitectónica de Aplicación Móvil	41
Fig. 12 Registro para Inyección de Dependencias.....	42
Fig. 13 Arquitectura de cada Caso de Uso en la Aplicación Móvil	43
Fig. 14 Inyección de Dependencias del Caso de Uso Notas	44
Fig. 15 Vista calendario actividades	47
Fig. 16 Detalles de actividad (Primera Versión).....	47
Fig. 17 Vista horario estudiante	48
Fig. 18 Vista detalles de horario estudiante.....	48
Fig. 19 Vista perfil estudiante con datos personales.....	49
Fig. 20 Vista perfil estudiante con credenciales.....	49
Fig. 21 Vista notas estudiante menú	50
Fig. 22 Vista notas estudiante ciclo actual	50
Fig. 23 Vista registro académico.....	54
Fig. 24 Vista actividades aula virtual	55
Fig. 25 Vista notas aula virtual	55
Fig. 26 Vista recursos aula virtual.....	59
Fig. 27 Vista de asistencia del estudiante.....	59
Fig. 28 Vista menú documentos estudiante.....	60
Fig. 29 Vista certificados de eventos	60
Fig. 30 Vista historial de certificados.....	61
Fig. 31 Vista formulario de solicitud de certificado.....	61
Fig. 32 Vista avance de asignatura	65
Fig. 33 Vista detalles asignatura versión final	65
Fig. 34 Vista descripción actividad	66
Fig. 35 Vista opciones de adjuntar archivos.....	66
Fig. 36 Test de normalidad del taller práctico	73

Fig. 37 Alfa de Cronbach encuesta SUS.....	74
Fig. 38 Rango resultado de Calidad en Uso.....	80

Índice de Tablas

Tabla 1 Trabajos Relacionados.....	5
Tabla 2 Roles del Proyecto.....	29
Tabla 3 Historia de Usuario 1.....	30
Tabla 4 Historia de Usuario 2.....	30
Tabla 5 Historia de Usuario 3.....	31
Tabla 6 Historia de Usuario 4.....	31
Tabla 7 Historia de Usuario 5.....	32
Tabla 8 Historia de Usuario 6.....	32
Tabla 9 Historia de Usuario 7.....	33
Tabla 10 Historia de Usuario 8	33
Tabla 11 Historia de Usuario 9	34
Tabla 12 Historia de Usuario 10	34
Tabla 13 Historia de Usuario 11	35
Tabla 14 Historia de Usuario 12	35
Tabla 15 Historia de Usuario 13	35
Tabla 16 Product Backlog	36
Tabla 17 Sprint Backlog - Sprint 1.....	45
Tabla 18 Revisión de criterios de aceptación Sprint 1.....	51
Tabla 19 Sprint Backlog - Sprint 2.....	53
Tabla 20 Revisión de criterios de aceptación Sprint 2.....	56
Tabla 21 Sprint Backlog - Sprint 3.....	57
Tabla 22 Revisión de criterios de aceptación Sprint 3.....	62
Tabla 23 Sprint Backlog - Sprint 4.....	64
Tabla 24 Revisión de criterios de aceptación Sprint 4.....	67
Tabla 25 Definición del modelo de calidad en uso.....	69
Tabla 26 Objetivos y tareas del taller práctico	70
Tabla 27 Preguntas encuesta SUS [75]	72
Tabla 28 Peso respuestas encuesta SUS.....	77
Tabla 29 Resultado SUS - Utilidad	77
Tabla 30 Resultados SUS - Comodidad.....	79
Tabla 31 Resultados Calidad en Uso.....	81

Resumen

Las plataformas móviles han tomado un rol fundamental en la provisión de servicios digitales, impactando significativamente la experiencia de los usuarios. En este contexto, la Universidad Técnica del Norte ofrece el servicio “Portafolio Estudiante” a través de una plataforma web desarrollada en APEX. Sin embargo, la falta de optimización para dispositivos móviles genera dificultades en el acceso y uso del sistema, afectando la satisfacción de los estudiantes.

Para abordar esta problemática, el presente trabajo implementó el servicio “Portafolio Estudiante” en una plataforma móvil, utilizando el framework Flutter para el desarrollo multiplataforma y adoptando una arquitectura de microservicios. La implementación siguió la metodología ágil SCRUM, permitiendo un desarrollo iterativo y adaptativo.

La evaluación del sistema se realizó con base en la norma ISO/IEC 25022, considerando métricas de eficacia, eficiencia y satisfacción. Los resultados indican que la aplicación móvil se encuentra dentro de un rango satisfactorio en calidad de uso, proporcionando funcionalidades clave como la consulta de horarios de clases, consulta de calificaciones, entrega de actividades y acceso a recursos educativos.

Abstract

Mobile platforms have taken a fundamental role in the provision of digital services, significantly impacting the user experience. In this context, the Universidad Técnica del Norte offers the “Portafolio Estudiante” service through a web platform developed in APEX. However, the lack of optimization for mobile devices generates difficulties in the access and use of the system, affecting student satisfaction.

To address this problem, the present work implemented the “Portafolio Estudiante” service on a mobile platform, using the Flutter framework for cross-platform development and adopting a microservices architecture. The implementation followed the agile SCRUM methodology, allowing an iterative and adaptive development.

The system was evaluated based on the ISO/IEC 25022 standard, considering effectiveness, efficiency and satisfaction metrics. The results indicate that the mobile application is within a satisfactory range in terms of quality of use, providing key functionalities such as class schedule consultation, grade consultation, delivery of activities and access to educational resources.

INTRODUCCIÓN

Planteamiento del Problema

Las plataformas móviles se han convertido en un canal esencial para consumir servicios digitales [1]. Esta tendencia también se ve reflejada a nivel nacional [2]. El cambio de paradigma en el consumo de servicios se ve principalmente marcado en los usuarios más jóvenes [3], lo que subraya la creciente importancia de ofrecer una experiencia de usuario optimizada para dispositivos móviles. Dentro de este contexto, es oportuno mencionar la encuesta realizada por Sosa en 2024 con el objetivo de conocer cuál es la percepción de los estudiantes sobre el servicio “Portafolio Estudiante” de la Universidad Técnica del Norte, en esta se observa que la plataforma web actual desarrollada en APEX presenta desafíos a los usuarios que necesitan o prefieren acceder a las funcionalidades académicas desde sus dispositivos móviles. Como se observa en la Fig. 1, la solución web no cuenta con interfaces que se adapten al tamaño de los dispositivos móviles. La encuesta señaló que los estudiantes se sienten insatisfechos con estas limitaciones, lo cual afecta negativamente la imagen de la institución dentro de la comunidad, al no cumplir con las expectativas modernas de accesibilidad digital. Con el fin de mitigar estas consecuencias negativas, se desarrolló una plataforma que permite realizar las funciones principales del servicio web a través de dispositivos móviles personales, lo que mejorara la usabilidad y accesibilidad del servicio.

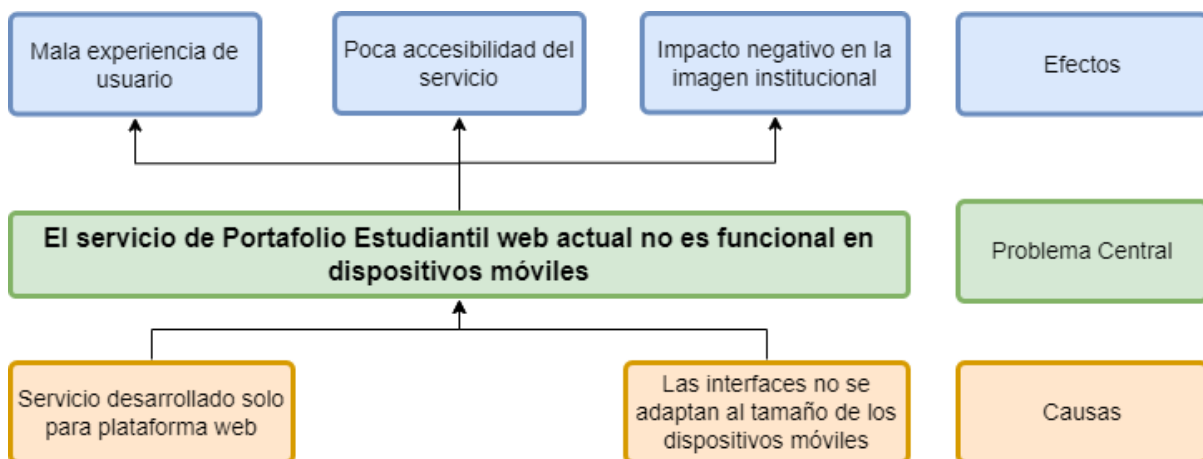


Fig. 1 Árbol de Problemas

Objetivos

Objetivo General

Implementar el 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

Objetivos Específicos

1. Establecer un marco conceptual para el estudio del desarrollo de una aplicación móvil de gestión académica.
2. Implementar una aplicación móvil para el servicio del "Portafolio Estudiante" utilizando el marco de trabajo SCRUM.
3. Evaluar la calidad en uso de la aplicación móvil para el servicio del "Portafolio Estudiante" con la norma ISO/IEC 25022.

Alcance

El alcance del presente trabajo fue desarrollar el servicio "Portafolio Estudiante" para la plataforma móvil de la Universidad Técnica del Norte, con el objetivo de proporcionar una solución eficiente y accesible que mejore la experiencia de los usuarios en dispositivos móviles. Para el desarrollo de la aplicación móvil se empleó una arquitectura limpia utilizando el framework multiplataforma Flutter. El aplicativo se implementó para las plataformas Android y iOS. Para el desarrollo backend se adoptó el uso de las tecnologías de preferencia de la Dirección de Desarrollo Tecnológico e Informático (DDTI) de la Universidad Técnica del Norte. Esta decisión se sustentó a través de la comparación entre APEX, Node, Python y Java, tecnologías elegidas por su rendimiento. El desarrollo se realizó siguiendo una arquitectura microservicios que permite una mejor organización del código y escalabilidad. La plataforma microservicios está conectada a la base de datos Oracle de la institución, como se muestra en la Fig. 2.

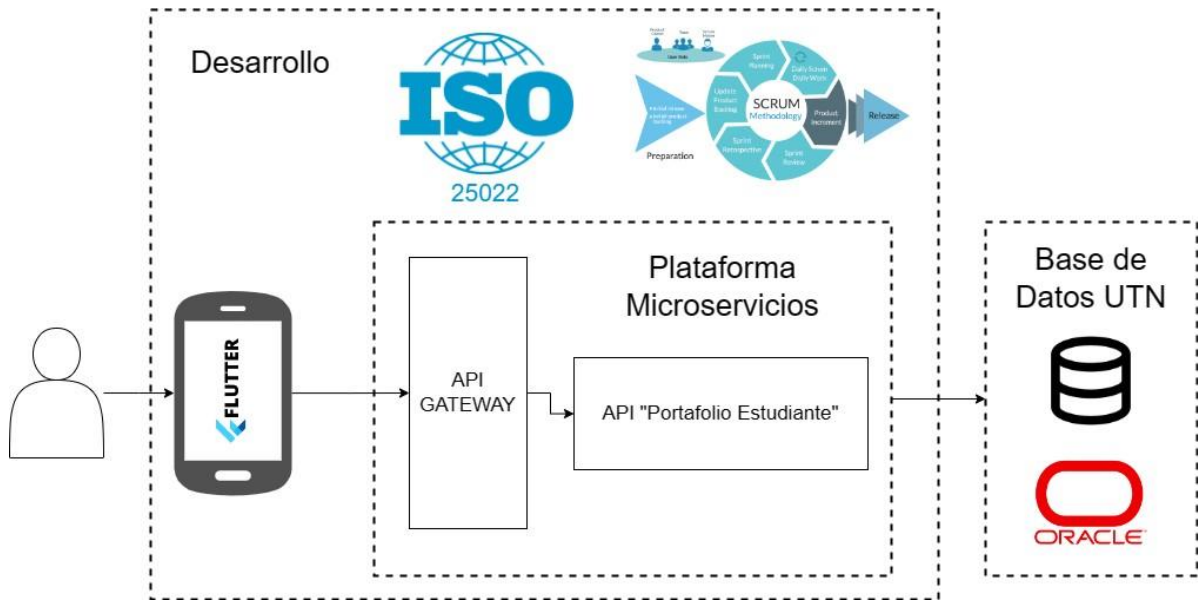


Fig. 2 Arquitectura Propuesta

Las funcionalidades desarrolladas dentro del servicio "Portafolio Estudiante" en la aplicación móvil son:

- **Calendario de Actividades:** Presenta plazos de entrega de actividades.
- **Acceso a recursos de las asignaturas:** El estudiante accede a los documentos o enlaces web que son cargados dentro de la asignatura.
- **Entrega de Actividades:** Facilita la entrega de tareas a través de dispositivos móviles, proporcionando una interfaz sencilla para adjuntar y enviar trabajos.
- **Revisión de Horarios del Estudiante:** Ofrece acceso a los horarios de clase.
- **Revisión de Notas del Estudiante:** Los estudiantes consultan sus calificaciones.
- **Revisión de los Syllabus:** Acceso a los syllabus de las asignaturas.
- **Perfil del estudiante:** Acceso a la información personal del estudiante y a los documentos cargados y generados en la plataforma.

- **Avance pragmático de las asignaturas:** El estudiante consulta el porcentaje de avance pragmático de la asignatura.

El proyecto se gestionó siguiendo el marco de trabajo SCRUM, que permite iteraciones rápidas, evaluación continua y adaptación a cambios [4]. Además, se evaluó la calidad en uso de la aplicación móvil siguiendo la norma ISO/IEC 25022 [5], como se muestra en la figura 2. Para esto se utilizarán las métricas de eficacia, eficiencia y satisfacción obtenidas de la norma ISO/IEC 25010 [6], [7].

Metodología

El presente trabajo de investigación es de tipo aplicada, pues se implementó el servicio “Portafolio Estudiante” en la plataforma móvil de la Universidad Técnica del Norte.

Se completaron las siguientes actividades relacionadas a cada objetivo:

1. Se realizó una investigación documental siguiendo el método Scope Review con el objetivo de recolectar información sobre el desarrollo de plataformas microservicios y aplicaciones móviles con el framework Flutter. Esta investigación utilizó las bases de datos científicas a las que la Universidad Técnica del Norte tiene acceso. A su vez, se empleó el gestor bibliográfico Mendely para llevar un registro de los trabajos encontrados.
2. Siguiendo la metodología SCRUM, se cumplieron las tres fases recomendadas. En el Pre-Juego se definieron los requisitos, en el Juego se realizó el desarrollo, y en el Post-Juego se llevó a cabo las pruebas de aceptación y entega [4], [7].
3. Una vez finalizado el desarrollo se procedió a realizar una evaluación de la calidad en uso del aplicativo utilizando la norma ISO/IEC 25022 a través de una encuesta y taller práctico con los usuarios del sistema [5], [7].

Justificación

Justificación Institucional y Tecnológica

La implementación del servicio "Portafolio Estudiante" en la plataforma móvil de la Universidad Técnica del Norte se alinea plenamente con el Objetivo Estratégico 4: "Fortalecer las capacidades institucionales" del Plan Estratégico 2021-2025 de la institución. Tomando como base el programa "Desarrollo de software" y su proyecto 4.22 "Desarrollo de servicios tecnológicos en APPs", esta iniciativa refuerza el compromiso institucional con la modernización tecnológica, posicionando a la universidad como una institución innovadora que responde de manera efectiva a los desafíos de la educación digital y mejora la experiencia académica de los estudiantes [8].

Justificación Social

Este proyecto se alinea con el Objetivo de Desarrollo Sostenible número 4, "Educación de calidad" [9]. Evidencia el compromiso de la Universidad Técnica del Norte con la mejora de la entrega de servicios educativos digitales. Los estudiantes se posicionan como los principales beneficiarios de esta innovación, pues se mejora su experiencia a través de un acceso más flexible a los servicios académicos.

Contexto de la investigación

Tabla 1 Trabajos Relacionados

INVESTIGACIÓN	APORTE
Tipo: Internacional An Android Application for Campus Information System [10].	En el artículo presentado se describe el desarrollo de una aplicación de gestión académica universitaria. Esta fue desarrollada únicamente para Android utilizando Android Studio. El presente trabajo tiene como aporte

	<p>el desarrollo un aplicativo de gestión académica multiplataforma utilizando Flutter.</p>
<p>Tipo: Local</p> <p>Desarrollo de una aplicación móvil utilizando el framework Flutter para fomentar el área turística del GAD de Pedro Moncayo [11].</p>	<p>La implementación de Flutter en el desarrollo de una aplicación móvil turística para el GAD demuestra la viabilidad de utilizar frameworks de desarrollo móvil multiplataforma para crear soluciones de alta calidad. El presente trabajo utilizó el framework mencionado para implementar una aplicación de gestión académica.</p>
<p>Tipo: Local</p> <p>Estudio de una arquitectura de microservicios mediante Spring Cloud para el desarrollo del módulo de registro y seguimiento médico de los deportistas en la Federación deportiva de Imbabura [12].</p>	<p>El trabajo mencionado enfatiza la efectividad de SCRUM dentro del desarrollo de una plataforma microservicios y una aplicación web. El aporte realizado por el presente trabajo es el desarrollo de un aplicativo móvil que se comunice con la plataforma microservicios utilizando SCRUM.</p>
<p>Tipo: Local</p> <p>Desarrollo de un sistema web para la automatización del proceso de mapeo sistemático de la literatura, y validado mediante un marco de trabajo de calidad de uso basado en las normas ISO/IEC 25000 para mejorar el proceso de investigación en los docentes de la Universidad Técnica del Norte [7].</p>	<p>El trabajo de grado mencionado realiza el proceso de validación a través de la familia de normas ISO/IEC 25000. Para esto utiliza como métricas la eficiencia, eficacia y satisfacción. El trabajo es utilizado como una guía referencial para definir el proceso de validación a seguir dentro del desarrollo del aplicativo móvil "Portafolio Estudiante" a través de la norma ISO/IEC 25022.</p>

<p>Tipo: Internacional</p> <p>Microservice-based projects in agile world: A structured interview [13].</p>	<p>El artículo científico menciona que después de realizar entrevistas a desarrolladores es correcto afirmar que la adopción de una arquitectura de software basada en microservicios es compatible con una metodología ágil. El presente trabajo implementó la metodología ágil SCRUM para el desarrollo no solo de microservicios, también de un aplicativo móvil.</p>
--	--

I MARCO TEÓRICO

1.1 Sistema Informático Integrado Universitario de la Universidad Técnica del Norte

1.1.1 Portafolio Estudiante UTN

El portafolio universitario se define, según el reglamento de la Universidad Técnica del Norte, como un conjunto de recursos para evidenciar y evaluar el aprendizaje de forma periódica con el fin de cumplir los objetivos planteados por la institución. Se establece que se debe acceder a este a través del portal web del sistema integrado universitario. El portafolio estudiante contiene el historial del estudiante durante su trayectoria universitaria a través de la recopilación de evidencia que refleja su desarrollo profesional. Entre esta información se encuentra el silabo de las asignaturas, horarios de clase, notas obtenidas, evidencia de actividades, entre otros [14].

1.1.2 Tecnologías utilizadas para el desarrollo

- *Oracle Database*

Oracle es conocido por ser el sistema de gestión de bases de datos de gran escala más utilizado en entornos empresariales. Se destaca por su eficiencia y robustez a la hora de realizar manejo de información de empresas de gran tamaño. Su estructura incluye componentes como segmentos y subsegmentos, que organizan lógicamente el espacio para objetos como tablas y bloques de datos, que son las unidades más pequeñas de almacenamiento [15].

Una de las herramientas que ofrece la base de datos Oracle es el lenguaje PL/SQL, este fue creado por Oracle en 1991 como un lenguaje procedural que extiende las funcionalidades del lenguaje SQL y las integra con su sistema de gestión. Este lenguaje permite ocultar información, encapsular datos, manejar errores, programación orientada a objetos, entre otras funcionalidades [16].

- *Oracle APEX*

Kvet define a Oracle Application Express (APEX) como una plataforma reconocida por ser líder en el desarrollo aplicaciones con cantidades mínimas de código. Proporciona soluciones robustas para crear sistemas escalables y de rápido desarrollo. La principal ventaja de esta herramienta es que permite el desarrollo de aplicaciones sin el conocimiento necesario de lenguajes de programación u otras tecnologías comunes dentro de la construcción de sistemas informáticos [17].

Oracle APEX fue creado en 2004 por Mike Hichwa con el nombre HTML DB, sin embargo, esto cambiaría en 2006 al actualizarse y tomar el nombre de Apliacation Express. Esta herramienta está integrada con Oracle Database lo que la excelente para el manejo de datos como validaciones y procesos de lógica de negocio. También es importante mencionar que la demanda de esta tecnología ha crecido con el tiempo. En el año 2022 sufrió un aumento repentino por lo cual lo hace una herramienta en demanda actual [18].

1.1.3 Otras plataformas universitarias similares en Ecuador

Estas plataformas facilitan la comunicación y organización entre estudiantes y docentes, permitiendo el acceso a contenidos educativos, el registro de calificaciones, y el seguimiento de actividades, tanto en la modalidad virtual como en el campus. La Universidad Central del Ecuador, por ejemplo, utiliza Moodle [19], una plataforma de código abierto ampliamente adoptada que ofrece flexibilidad en el diseño de cursos y la administración de actividades presenciales, permitiendo a los docentes estructurar sus clases de manera organizada y accesible para los estudiantes, sin importar el formato [20].

La Universidad Espíritu Santo [21] ha elegido Blackboard como su plataforma principal. Blackboard es reconocida no solo por sus funciones en entornos virtuales, sino también por su robustez para gestionar cursos presenciales, proporcionando herramientas de evaluación, foros de discusión y colaboración que enriquecen la experiencia en el aula. Estas plataformas permiten a los estudiantes acceder a recursos adicionales y a materiales de clase desde cualquier lugar, complementando las sesiones presenciales con actividades en línea, lo cual impulsa una

experiencia académica integral y acorde con las exigencias actuales de calidad y accesibilidad en la educación superior [22].

1.2 Aplicaciones Móviles

A diferencia de programas tradicionales que están diseñados para ser ejecutados en PC, laptops o en el navegador, las aplicaciones móviles son software especialmente desarrollado para teléfonos inteligentes o tabletas. Durante su construcción se consideran las características particulares de estos dispositivos como la pantalla táctil, cámara, sensores y dispositivos de conectividad para brindar una mejor experiencia de uso y funcionalidad [23].

1.2.1 Tendencia de los usuarios hacia el uso de plataformas móviles

En 2023 se publicó un estudio en Bangladesh donde se afirma que el uso de teléfonos inteligentes ha afectado varios aspectos de la vida. El uso de este permite a la población permanecer comunicada con más frecuencia mejorando así su vida social. También mejora la calidad del estudio, pues permite a los estudiantes acceder a información de forma más sencilla, a la vez que se mejora la comunicación con docentes y compañeros de clase [24]. De igual forma otro estudio realizado en Estados Unidos muestra que el uso de teléfonos inteligentes y tabletas está principalmente marcado dentro de la población con edades entre 18 y 29 años. Esto ocasiona que cada vez es más frecuente su uso en universidades y otro tipo de centros educativos. Este uso no es únicamente en tiempos libres, pues también se hace uso durante la clase [25].

A nivel nacional también se observa un claro aumento en el uso de dispositivos móviles, en 2014 el 13.3% de la población superior a 5 años tenía acceso a este dispositivo, esta métrica ha aumentado en 2023 a 55.6% siendo un total de 9.4 millones de personas. Esta distribución es equitativa entre hombres y mujeres lo cual significa que ambos sexos tienen acceso a esta herramienta. Sin embargo, si existe una distinción en el área de ubicación pues en la zona urbana el 63.1% usa este medio de comunicación, mientras que en la zona rural solo el 39.4% lo usan [26].

1.2.2 Aplicaciones en entornos Universitarios

En Ecuador, el desarrollo de aplicaciones móviles para universidades es limitado. La mayoría de las aplicaciones disponibles en PlayStore y AppleStore se enfocan en ofrecer funcionalidades como la consulta de horarios, calificaciones y noticias institucionales. Entre estas plataformas se encuentran “UDLA+” aplicación de la Universidad de las Américas [27], “UDA” aplicación de la Universidad del Azuay [28], “PUCE APP” aplicación de la Universidad Católica de Quito [29], entre otras.

Pocas de estas aplicaciones ofrecen herramientas avanzadas que faciliten una interacción integral entre estudiantes y profesores. Por ejemplo, la posibilidad de subir tareas, generar documentos institucionales o gestionar el avance académico en tiempo real desde el celular es limitada en muchas plataformas. Es importante mencionar que también existe un descontento general en estas plataformas, esto se observa a través de las reseñas negativas que se encuentran en la tienda de aplicaciones [27], [28], [29].

A su vez, se debe tomar en cuenta la existencia de aplicaciones con buenas reseñas como “UEES” aplicación de la Universidad Espíritu Santo. Esta aplicación permite realizar acciones como solicitar exámenes de recuperación, certificados de notas, control de saldo en tarjetas institucionales, entre otras actividades que facilitan la interacción con los estudiantes [30].

1.2.3 Aplicaciones móviles multiplataforma

Según la documentación oficial de Kotlin, un framework para la implementación móvil multiplataforma, este término se define como un enfoque que permite desarrollar una única aplicación que sea capaz de ejecutarse en varios sistemas operativos móviles. Esto significa que el código se comparte y permite construir la misma aplicación para Android y iOS. Gracias a este método el desarrollo de aplicaciones se reduce en tiempo y costo [31]. Entre los frameworks más famosos para el desarrollo de este tipo de plataforma se encuentran React Native y Flutter [32].

1.2.4 Lenguaje de programación Dart

Hassan define a Dart como un lenguaje de código abierto con propósito general desarrollado por Google, que utiliza completamente el paradigma de programación orientada a objetos. Este lenguaje oculta los detalles internos de las nuevas plataformas de desarrollo, permitiendo a los desarrolladores enfocarse en la lógica de negocio, a la vez que el lenguaje se encarga de hacer uso de las características innovadoras de estas plataformas principalmente orientadas a la optimización. Dart tiene una sintaxis sencilla y clara que permite crear aplicaciones escalables. Una de sus características más destacadas es que permite a los usuarios desarrollar aplicaciones concurrentes con ejecución paralela mediante su modelo basado en isolates [33].

1.2.5 Desarrollo de aplicaciones móviles multiplataforma con Flutter

Flutter es un framework desarrollado por Google que utiliza el lenguaje de programación Dart. Esta herramienta provee una solución unificada para desarrollar, a través de una misma base de código, aplicaciones que se ejecuten en plataformas móviles como teléfonos inteligentes o tabletas, así como también es laptops y computadoras de escritorio. Una de las características más reconocidas de este framework es la capacidad de realizar “hot reload” lo que permite implementar cambios de forma rápida durante el desarrollo [34].

1.2.6 Arquitectura limpia en Flutter

La arquitectura limpia también conocida como Clean Architecture es una arquitectura de diseño de software que sigue los principios de código limpio SOLID. Este es el resultado de la combinación de buenas prácticas generadas a lo largo del tiempo. En esta se prioriza la lógica de negocio sobre los demás componentes, a la vez que se trata de reducir al mínimo la dependencia entre estas piezas de software. Esta forma de estructura del código fue divulgada por Robert Martin. El propone separa 4 capas como se muestra en la Fig. 3, la capa de entidades, la capa de casos de uso, la capa de adaptadores o interfaces y finalmente la capa del framework o dispositivos. Estas se agrupan de forma jerárquica en un círculo. Las capas internas no tienen conocimiento de las capas más externas [35].

Este concepto de arquitectura limpia puede ser aplicado de diferentes formas dependiendo del lenguaje o framework a utilizar, por ejemplo, Boukhary y Colmenares proponen realizar una separación de 3 capas cuando se va a utilizar Flutter. La capa de infraestructura que se encarga del acceso a datos y otros medios, la capa de aplicación encargada de interactuar con el framework y finalmente la capa de dominio que contiene la lógica del negocio y debe ser totalmente independiente [37].

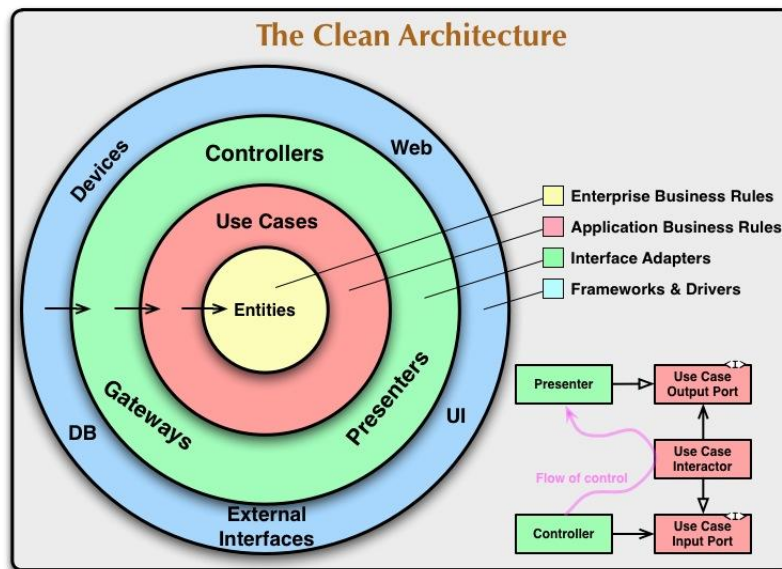


Fig. 3 Diagrama de Arquitectura Limpia [36]

Para hacer uso de esta arquitectura en Flutter es posible usar la librería "get_it", que simplifica la conexión entre componentes a través de la inyección de dependencias. Esta librería pone a disposición del desarrollador un localizador de servicios que permite separar interfaces de su implementación, a la vez que permite acceder a estas implementaciones desde cualquier parte del código siguiendo normas definidas previamente, como crear un único objeto que será compartido para cada petición que se haga al localizador de servicios [38].

- *Capa de Dominio*

Esta capa se enfoca en definir las entidades las cuales encapsulan lógica de negocio, estas pueden ser objetos que se encuentran estrechamente relacionados con la información

almacenada en la base de datos. Las modificaciones de las capas externas no generan modificaciones a esta capa [39]. Estas se encuentran en el nivel más alto de abstracción y no pueden interactuar con las capas externas. Las entidades en el mismo nivel pueden interactuar entre si a través de interfaces [35].

En esta capa también se definen los casos de uso, estos son las reglas de negocio que definen la existencia del aplicativo desarrollado. Controlan el flujo de la información con el fin de cumplir el su propósito específico [35]. Estos componentes interactúan con los datos a través de Repositorios, los cuales son definiciones de interfaces que posteriormente serán implementadas en otra capa. Al igual que las entidades, modificaciones en la capa externa no influyen en su comportamiento [37].

- *Capa de Infraestructura*

En esta capa se implementan las interfaces definidas en la capa de dominio. Estas interactúan con los dispositivos o medios externos a la aplicación como una base de datos local, un API, cámara del dispositivo, entre otros. Una interfaz puede ser implementada de distintas maneras debido a que la capa de dominio depende de una interfaz y no de una implementación específica [35]. En esta capa se puede hacer uso de librerías para realizar peticiones http para consumir APIs como DIO, la cual se destaca por ser un paquete que soporta configuración global, interceptores, carga y descarga de archivos, entre otras características [40]. También se puede usar librerías que permitan la creación de base de datos locales como Floor, esta provee una abstracción de SQLite que permite realizar un mapeo automático entre entidades cargadas a memoria y los registros graduados en la base de datos [41].

- *Capa de Presentación*

Esta capa es la más dependiente del framework pues se encarga de presentar el contenido a los usuarios a través de los mecanismos que Flutter define [37]. Para presentar información Flutter desarrollo el concepto de *widget* y su unión que forma el *widget tree*. Un *widget* es una clase de dart que describen una vista para posteriormente será renderizada por el motor de flutter. En otros framework se conoce a este concepto con el nombre de *componente*. Por defecto

flutter contiene varios *widgets* que permiten la creación de la aplicación como *layouts, styles, animations, positioning, alignment*, entre otros. Es trabajo del desarrollador crea nuevos *widgets* que cumplan con las necesidades específicas de la aplicación [42].

1.3 Arquitecturas Microservicios

1.3.1 Interfaz de programación de aplicaciones (API)

Se define una API como interfaz que comunica piezas de software sin la interacción humana. Estas exponen servicios de negocio a través de *endpoints*, estas rutas son utilizadas por los desarrolladores que construyen otras aplicaciones que deseen comunicarse con la API. Para esto se especifica un contrato que define los formatos de entradas y salidas de cada uno de los *endpoints*. Es importante destacar que una API se puede acceder desde distintos orígenes como teléfonos inteligentes, tabletas, consolas de videojuegos, navegador web entre otros [43].

1.3.2 Plataformas microservicios

Un estudio realizado el 2024 define a las plataformas microservicios como una arquitectura de software en la cual un sistema grande y monolítico se divide en múltiples servicios pequeños e independientes llamados microservicios, que se comunican entre si a través de APIs. Esta arquitectura mejora la mantenibilidad, escalabilidad y tolerancia a fallos gracias a su acoplamiento débil. Otro factor importante es que, debido a la autonomía de cada servicio, estos pueden evolucionar de forma independiente, sin embargo, es importante mantener en consideración que todos los servicios que dependen entre si se deben mantener de forma conjunta para asegurar compatibilidad [44].

1.3.1 Selección del Lenguaje de programación

Para fundamentar la elección de tecnologías, se llevó a cabo un proceso de evaluación riguroso que combinó una revisión sistemática de la literatura (SRL) y el análisis de encuestas reconocidas en el ámbito profesional como Stack Overflow Developer Survey 2024 [45] y la encuesta IEEE Spectrum 2024 [46].

Con base en los hallazgos de la SRL y las encuestas, se seleccionaron cuatro combinaciones de lenguajes y frameworks para realizar un experimento comparativo:

- JavaScript/TypeScript con el framework NestJS.
- Java con Spring Boot.
- Python con Flask.
- SQL con APEX.

El experimento consistió en realizar consultas a una base de datos, aumentando progresivamente la complejidad mediante la profundidad de los joins y el volumen de registros. Las consultas se diseñaron con las siguientes características:

- Complejidad: Se realizaron consultas con niveles de profundidad de 1, 2, 3 y 4 joins anidados.
- Volumen de Datos: Cada nivel de consulta se ejecutó con conjuntos de datos de 1, 10, 100, 1,000, 10,000 y 100,000 registros.

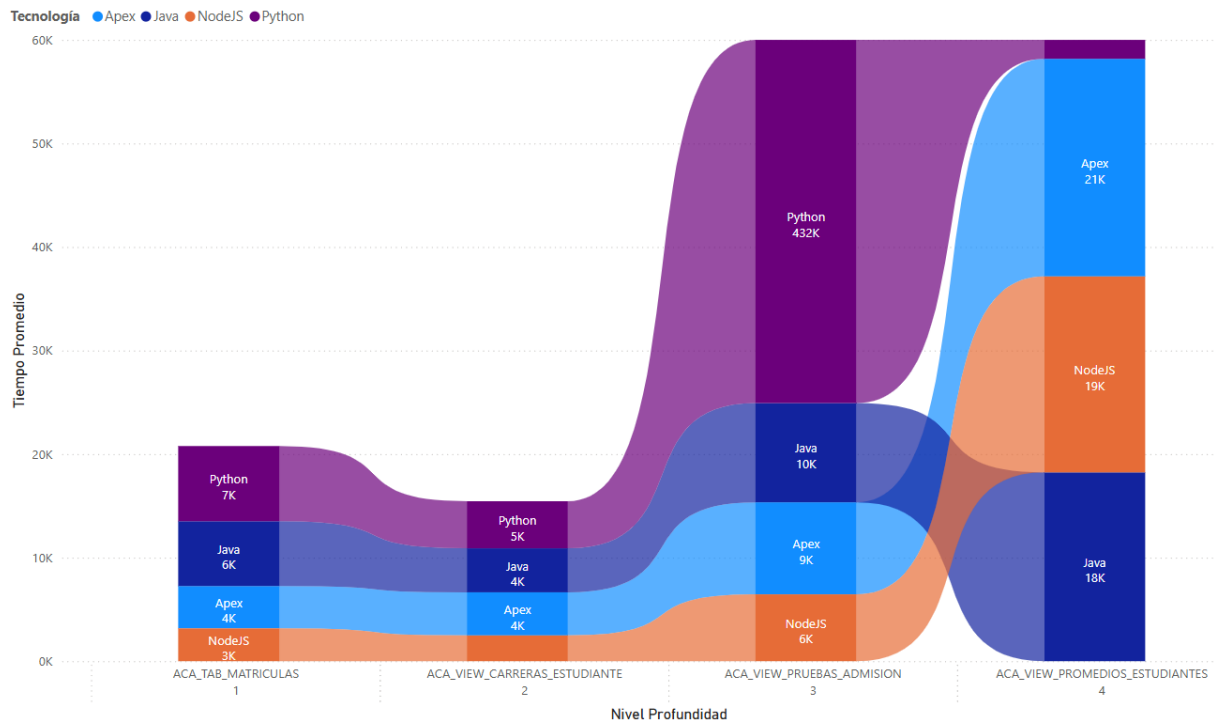


Fig. 4 Resultados comparación tecnologías

El objetivo del experimento fue medir el rendimiento de cada combinación en términos de tiempo de ejecución de las consultas, considerando escenarios de creciente complejidad. Los resultados del experimento mostraron que JavaScript/TypeScript con NestJS mantuvieron un rendimiento superior en términos de velocidad de ejecución mientras se aumentaba la complejidad como se muestra en la Fig. 4. Esto posicionó a NestJS como la opción destacada para el desarrollo del presente proyecto.

1.3.2 Lenguaje de programación TypeScript

Se define a TypeScript como una extensión del lenguaje de programación web popular JavaScript. Este facilita el desarrollo de aplicaciones escalables de gran tamaño al introducir un sistema de tipos que incluye clases, interfaces entre otras funcionalidades [47]. Todo código JavaScript es código TypeScript válido, de esta forma se facilita la transición entre lenguajes para desarrolladores que no tengan experiencia sólida con la herramienta [48]. Finalmente, es importante conocer que TypeScript se traduce a código JavaScript antes de ejecutarse, por lo cual el rendimiento de los dos lenguajes es el mismo y únicamente puede variar por las optimizaciones hechas por el “transpilador” o aquellas que se agregan de forma manual a cada lenguaje [49].

1.3.3 Desarrollo de aplicación microservicios con NestJS

NestJS es un framework de Node.js desarrollado con el fin de construir aplicaciones backend escalables y fáciles de mantener. Se diseñó con el objetivo de utilizar TypeScript pues combina elementos de la programación orientada a objetos, programación funcional y programación reactiva, sin embargo, es posible utilizar JavaScript puro si los desarrolladores así lo desean [50]. De forma interna este framework utiliza express como librería que permite crear servidores HTTP, pero también es posible configurar otras librerías para este propósito como Fastify [51].

- *Controlador*

Esta pieza de software se encarga de manejar todas las peticiones http y enviar las respuestas al cliente. Las peticiones llegan al controlador adecuado a través de un mecanismo de enrutamiento. Para crear un controlador es indispensable señalar el método HTTP que este

cumplirá, estos pueden ser: GET, POST, PUT, DELETE. En NestJS se los identifica a través de una anotación con el símbolo @ antes del nombre del método [52].

- *Servicio*

Objeto encargado de gestionar la lógica del negocio, para esto es posible comunicar múltiples servicios con el fin de llevar a cabo una funcionalidad [52]. Esta comunicación entre servicios y con el controlador respectivo se realiza a través de la inyección de dependencias por constructor. Este patrón de diseño es útil cuando se desea generar código desacoplado, pues cada clase recibe todas sus dependencias a través del constructor y no hace uso directo de clases concretas, sino a través de interfaces, permitiendo así intercambiar piezas de código de forma sencilla [53].

- *Middleware*

Se define como una pieza de código que se encarga de conectar dos capas de software. Por ejemplo, es posible ubicar un middleware entre la capa de servicio y la capa de aplicación [54]. En NestJS los middlewares se utilizan para ejecutar código antes de que una petición sea recibida por su controlador respectivo [52].

1.4 Metodología Scrum

En 1990 nacieron las metodologías ágiles cuando un grupo de ingenieros de software concluyeron que era imposible realizar entregas de productos de software a tiempo con las metodologías del momento [55]. Para esto se escribió el manifiesto ágil que consta de 4 principios y 12 valores que guían el desarrollo ágil de software [56]. Actualmente Scrum es la metodología líder dentro del agilismo debido al valor que esta genera para los clientes [55]. Scrum es un marco de trabajo desarrollado en 1993 derivado de la metodología ágil y el empirismo. Este fue propuesto por Ken Schwaber y Jeff Shutherland al inspirarse en el deporte Rugby que se enfoca en la estructura y organización durante el juego [57].

Según Huertas, en su trabajo publicado en 2024, Scrum es un marco de trabajo ágil que incentiva la colaboración entre los miembros del equipo para llevar a cabo trabajos complejos y

adaptativos. Se implementa un enfoque interactivo e incremental a través de marcos de tiempo llamados *sprint*, este enfoque se utiliza para reducir la incertidumbre y mitigar riesgos a lo largo del tiempo. Es importante mencionar los 3 pilares empíricos de scrum que se deben seguir a lo largo de las 3 fases de scrum [58]:

- I. **Transparencia:** El proceso de trabajo debe ser visible para todos los interesados del producto permitiendo así la inspección.
- II. **Inspección:** Todos los artefactos de scrum y el progreso del trabajo deben ser inspeccionados con regularidad para detectar problemas o posibles cambios.
- III. **Adaptación:** Se requiere que el equipo scrum se adapte en el momento que se descubra algo nuevo.

1.4.1 Fase 1. Pre-Juego

En esta fase se define la visión inicial del proyecto que puede ser mejorada a lo largo del mismo. Se prepara un plan con el tiempo, costo de entrega y numero de iteraciones o *sprints* hasta la entrega final. También se llevan a cabo tareas como la identificación y evaluación de riesgos, validación de las herramientas de desarrollo y la definición del equipo de trabajo [58].

- *Artefactos de SCRUM*

Los artefactos de Scrum representan trabajo o valor. Son usados con el fin de maximizar la transparencia. La guía de Scrum especifica 3 artefactos [59]:

- I. **Product Backlog:** Es una lista ordenada de lo que se necesita agregar al producto. Estos elementos son estimados por las personas que realizan el trabajo.
- II. **Sprint Backlog:** Es el plan de trabajo para todo el sprint, este debe ser actualizado constantemente cada vez que se aprende algo nuevo.
- III. **Increment:** Es un paso que acerca al equipo de trabajo a cumplir con el objetivo del producto. Dentro de un *sprint* se pueden llevar a cabo varios *increments*.

Estos artefactos son referenciados a lo largo de toda la guía de scrum, pues son la base para entender cómo funciona este marco de trabajo ágil. Cada miembro del equipo Scrum crea o administra estos artefactos, asegurando así los 3 pilares de la metodología que mencionados anteriormente.

- *Roles de SCRUM*

Scrum define 3 roles fundamentales que se encuentran sin jerarquía alguna:

- I. **Dueño del Producto (Product Owner):** Se encarga de maximizar el trabajo generado del equipo de desarrollo. Es el único encargado de administrar el Producto Backlog a través de ordenar los elementos de la lista, expresar los elementos de la lista a todos los demás roles para que estos los comprendan [60]. Este debe contar con buenas habilidades de comunicación debido a que son los principales intermediarios entre los otros roles de scrum y los interesados del producto [58].
- II. **Equipo de Desarrollo (Development Team):** Esta conformado por los encargados de generar los incrementos del proyecto. Son los responsables de crear un plan para el sprint y garantizar la culminación de estos elementos a través del cumplimiento de la definición de hecho de cada uno [58].
- III. **Scrum Master:** Es el responsable de llevar el trabajo de acuerdo a la guía de Scrum, para esto debe asegurar que todos los demás roles entiendan la metodología. Es el responsable de brindar asistencia a los demás roles y a la organización [55].

1.4.2 Fase 2. Juego

En esta fase se realizan todos los *sprints* necesarios para culminar el proyecto. Usualmente es un periodo de tiempo de 4 semanas en el que se realizan actividades con el fin de generar incrementos. Dentro de este periodo también se llevan a cabo 4 eventos definidos por Scrum:

- I. **Sprint Planning:** En esta reunión que incluye a todos los roles de Scrum en donde se establece el trabajo que se llevara a cabo durante todo el *sprint* [59]. Para esto el equipo de desarrollo define lo que puede terminar en el tiempo establecido y el dueño del

producto verifica cual es el objetivo del *sprint*. Se recomienda utilizar como máximo 8 horas de trabajo para iteraciones de 4 semanas [58].

- II. **Daily Scrum:** Es un evento con una duración de 15 minutos en donde el equipo de desarrollo inspecciona el avance hacia el objetivo del *sprint*. Sirve para mejorar la comunicación entre los miembros del equipo, identificar posibles problemas y tomar decisiones con base a los nuevos hallazgos [59].
- III. **Sprint Review:** El objetivo de este evento es inspeccionar el resultado del *sprint* a los interesados. Estos pueden tomar decisiones para adaptarse a nuevo aprendizaje adquirido durante el tiempo o satisfacer nuevas necesidades. Esta reunión debe tardar como máximo 4 horas para iteraciones de 4 semanas [59].
- IV. **Sprint Retrospective:** El principal objetivo de este evento es planificar como mejorar con respecto al *sprint* pasado. Esta reunión entre todo el equipo Scrum no debe tardar más de 3 horas para iteraciones de 4 semanas [59].

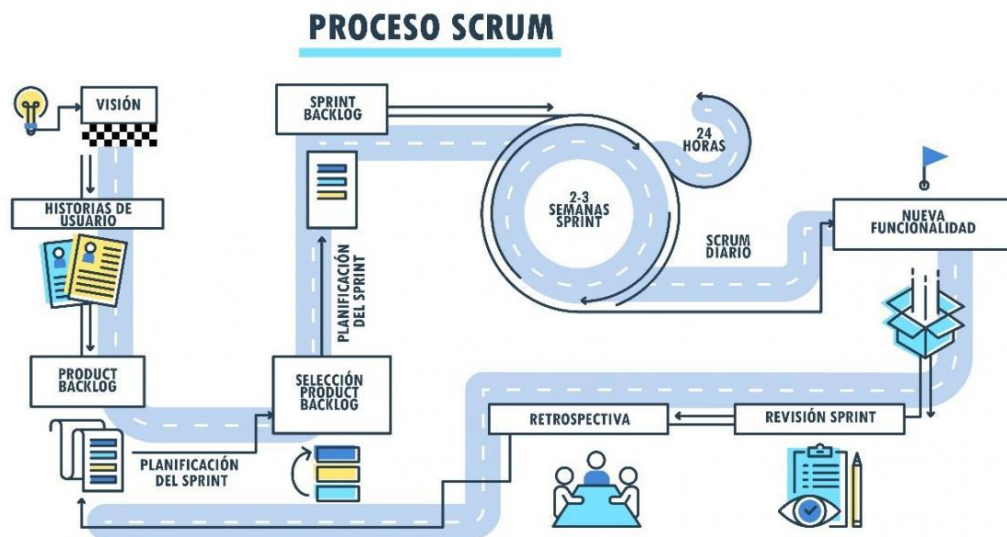


Fig. 5. *Proceso de Scrum* [61]

Juntos estos 4 eventos y los artefactos mencionados anteriormente forman el núcleo de Scrum como se muestra en la Fig. 5.

1.4.3 Fase 3. Post-Juego

En esta fase, se llevan a cabo pruebas de integración finales para asegurar que todos los componentes del sistema funcionen de manera cohesiva y sin errores antes de la entrega del producto a los clientes. Además, se crea manuales de usuario que guiarán a los usuarios en la utilización del sistema. También se desarrolla material de apoyo destinado a facilitar las capacitaciones realizadas por parte del equipo de desarrollo a los usuarios del sistema [58].

1.5 Modelo de evaluación de calidad

La calidad de software tiene muchas definiciones, Gao la define como el grado en el que un producto de software cumple los requerimientos deseados bajo ciertas circunstancias. Un modelo de calidad es un conjunto de características o métricas que conforman la base para medir y predecir la calidad del software. En la actualidad el modelo de calidad más utilizado es el que ofrece la familia de estándares ISO/IEC 25000 [62] el cual se encuentra dividido en varias secciones como se muestra en la Fig. 6.



Fig. 6 División de la norma ISO/IEC 25000 [63]

1.5.1 Norma ISO/IEC 25010

La norma ISO/IEC 25010 establece un conjunto de características esenciales para evaluar la calidad de un producto de software. Esta norma consta de 8 características principales que incluyen funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad, compatibilidad y seguridad. Cada una de estas áreas se subdivide en aspectos más detallados que permiten una evaluación minuciosa de diferentes facetas del software, asegurando que cumpla con los requisitos de calidad necesarios [64].

1.5.2 Calidad en Uso

Kurosu define a la calidad en uso como la calidad de un producto de software desde el punto de vista del usuario. Es importante mencionar que este concepto depende del contexto del usuario y se debe considerar su entorno o sus conocimientos técnicos [65]. Otra definición a este concepto se encuentra en el estándar ISO/IEC 25000. Este define a la calidad en uso como el grado en que un producto de software puede ser usado por usuarios específicos para cumplir sus objetivos, tomando en cuenta la libertad ante riesgos, eficiencia, efectividad y satisfacción con la que las actividades se llevan a cabo [66].

1.5.3 Norma ISO/IEC 25022

Esta norma define cuales son las métricas necesarias para medir la calidad en uso de las características definidas en la norma ISO/IEC 25010. También contiene explicaciones de cómo se miden estas métricas [67]. Entra las características mencionadas en este estándar se encuentran las siguientes [68]:

- *Eficacia*

Mide la capacidad del sistema para que los usuarios logren sus objetivos correctamente en un contexto de uso específico.

Algunas de sus sub características son:

1. Tareas completas: Evalúa el porcentaje de tareas que los usuarios logran completar satisfactoriamente.
2. Objetivos logrados: Mide el grado en el que los objetivos planteados por los usuarios han sido alcanzados mediante el sistema.
3. Errores en tareas: Registra la cantidad de errores cometidos durante la realización de tareas específicas.

- *Eficiencia*

Mide la cantidad de recursos gastados en comparación con la eficacia obtenida.

Algunas de sus sub características son:

1. Tiempo de tareas: Mide el tiempo necesario para completar una tarea específica.
2. Eficiencia del tiempo: Mide el tiempo necesario para completar un objetivo.

- *Satisfacción*

Mide el grado en que las necesidades de los usuarios han sido cubiertas.

Algunas de sus sub características son:

1. Utilidad: Representa el grado en que los usuarios consideran que el sistema satisface sus necesidades.
2. Confianza: Evalúa si los usuarios sienten que el sistema es confiable y seguro para usar.
3. Comodidad: Mide la percepción del usuario sobre la facilidad de uso y la ausencia de molestias al interactuar con el sistema.

1.5.4 Escala de medición de Calidad en Uso

En el presenta trabajo se tomó en cuenta la escala de medición que se propone en la ISO/IEC 25040. Esta permite evaluar de forma cuantitativa los resultados obtenidos al aplicar las métricas descritas en la anterior sección. La escala se divide en dos categorías y cada una esta a su vez está dividida en 2 subcategorías como se muestra en la Fig. 7.

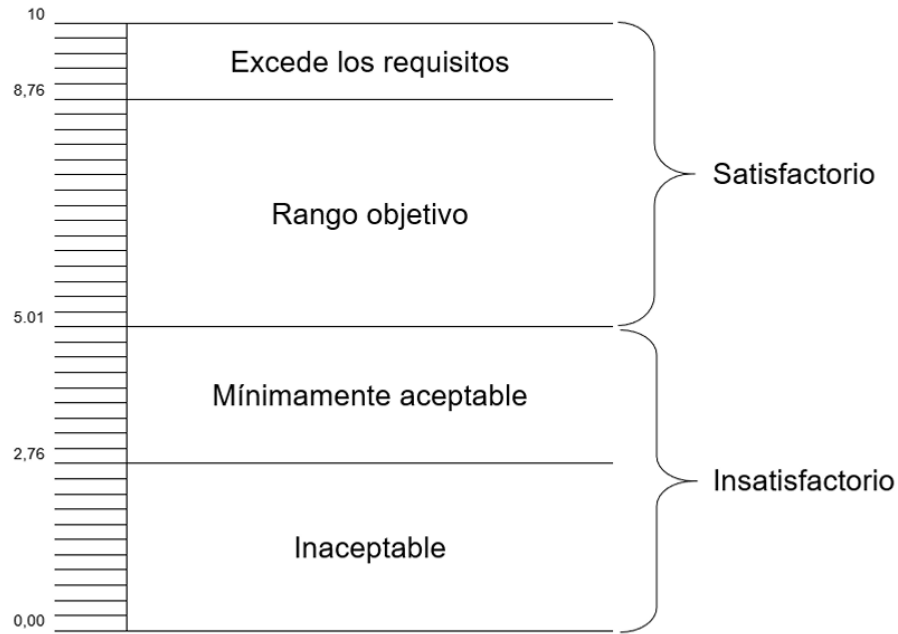


Fig. 7 Escala de medición de Calidad en Uso [69]

1.6 Trabajos Relacionados

Dentro del contexto internacional se puede mencionar el trabajo publicado en 2020 por Reetu Malhorta. Este tuvo como objetivo el desarrollo de una aplicación móvil para gestionar la información del campus universitario RIMT de la India. Para la construcción de la aplicación se empleó la tecnología Android Studio y la base de datos SQLite. Durante las pruebas de funcionamiento se utilizó el framework de código abierto robotium y se midió el rendimiento por medio de la tecnología promethee. La aplicación resultó ser efectiva para mejorar el manejo de información de los estudiantes y se consideró como un paso adelante para la construcción de un campus inteligente debido al gran incremento de usuarios (18%) en el último año. Finalmente, es oportuno mencionar que esta aplicación únicamente fue desarrollada para la plataforma Android. Debido a esto otras plataformas como iOS se quedaron fuera del alcance del proyecto [70].

En enero de 2024 Hüseyin Ünlü publicó su trabajo con el objetivo de medir cómo el uso de una metodología ágil influye a lo largo de las fases de la construcción de software orientado a microservicios. Para esto se entrevistó a 20 desarrolladores de organizaciones y países distintos,

recolectando experiencias positivas y negativas sobre los retos que tuvieron que superar. Los resultados obtenidos demuestran que este tipo de metodologías son compatibles con el desarrollo de microservicios. El 90% de las respuestas afirma utilizar métodos ágiles como historias de usuario para la recolección de requerimientos. También el 60% de los encuestados respondió utilizar Jira como herramienta para llevar el control de la metodología utilizada. La principal limitación del trabajo es que se enfoca en recolectar un punto de vista general de las experiencias previas de las organizaciones, sin concentrarse en una sola metodología ágil como Scrum. Además, no se presenta un caso concreto en el que se apliquen las variables de estudio [71].

Dentro del contexto local se puede mencionar al trabajo de grado realizado por Lescano Iván. En este se desarrolló una aplicación móvil para fomentar el área turística del GAD de Pedro Moncayo. Las tecnologías utilizadas fueron Flutter debido a su capacidad de crear aplicaciones multiplataforma. También se utilizó Node.js y Mongo DB en el backend gracias a su versatilidad. Finalmente se utilizó la metodología de desarrollo Scrum y se validó los resultados a través de la norma ISO/IEC 25000. Después de aplicar la encuesta SUS a 32 personas se obtuvo un puntaje de 81.79 superando a la media que es 68. Este puntaje demuestra que la aplicación tiene un grado de aceptación A, pues este cumple con las expectativas de los usuarios. La limitación principal de este trabajo es que únicamente se consideró una arquitectura monolítica debido al nivel de complejidad que supone la implementación de una arquitectura orientada a microservicios [11].

En 2020 Flores Jeferson publicó su trabajo de grado sobre el estudio de una arquitectura de microservicios para el desarrollo del módulo de registro y seguimiento médico de los deportistas en la federación deportiva de Imbabura. Para esto se utilizó la tecnología Spring Cloud debido a la facilidad de desarrollo que ofrece en plataformas microservicios. También se utilizó el framework Angular 4 debido a su gran flexibilidad en el desarrollo frontend y la base de datos PostgreSQL que ha demostrado ser confiable en proyectos de esta magnitud. El desarrollo fue guiado por la metodología Ágil Scrum y se validó a través de la norma ISO/IEC 25000. Los resultados de la calidad en uso de la aplicación se midieron a través de dos métricas aplicadas a 20 usuarios, en eficiencia se obtuvo un puntaje de 83.04 y en satisfacción 78.68 obteniendo así

un puntaje final de 80.86 el cual refleja que la aplicación se encuentra dentro de un rango satisfactorio. Finalmente, se debe mencionar que el estudio no consideró la implementación de una aplicación móvil sobre una aplicación web que cumpla con las mismas funcionalidades, pero se adapte mejor a la tendencia actual de uso de dispositivos móviles [12].

Bastidas Edwin publicó en 2020 un trabajo sobre el desarrollo de un sistema web para la automatización del proceso de mapeo sistemático de literatura con el fin de mejorar el proceso de investigación de docentes de la universidad Técnica del Norte. Para esto utilizó Node.js junto a su librería Express.js que permite el desarrollo rápido y flexible de una API. También se utilizó la tecnología GraphQL para servir los datos al cliente y Neo4j para almacenarlos. En el cliente se utilizó React.js y Bootstrap para un desarrollo rápido de componentes visuales. Para medir los resultados se realizó un taller práctico con usuarios reales del sistema y llevó a cabo la encuesta SUS. El resultado final obtenido (8.41) demuestra que la aplicación se encuentra dentro de un rango satisfactorio en la escala de medición de la calidad en uso. La principal limitación de este trabajo es la omisión del uso de un framework robusto para la construcción del backend como Nest.js que facilita la implementación de todas las funcionalidades manteniendo una estructura limpia y consistente dentro del código [7].

1.7 Resultados obtenidos del estudio

La Universidad Técnica del Norte enfrentó la necesidad de implementar tecnologías innovadoras que respondan a las expectativas y demandas de los estudiantes en relación con el servicio de Portafolio Estudiante. Este servicio juega un papel fundamental en la gestión académica y administrativa, por lo que es esencial que se adapte a las tendencias tecnológicas actuales para ofrecer una experiencia más eficiente y accesible.

En este contexto, las aplicaciones móviles se presentan como una solución ideal. Estas plataformas no solo se alinean con los hábitos tecnológicos de los estudiantes, sino que también ofrecen una forma práctica y conveniente de interactuar con los servicios académicos. Para el desarrollo de esta aplicación móvil, se identificó a Flutter como una tecnología clave. Su capacidad para crear aplicaciones multiplataforma permite desarrollar código que funcione tanto en

Android como en iOS, lo cual optimiza los recursos y reduce significativamente los tiempos de desarrollo. Además, su flexibilidad y robustez facilitan la implementación de características avanzadas y la integración con otras tecnologías.

A nivel de backend, se seleccionó una arquitectura de microservicios. Este enfoque no solo facilita la independencia evolutiva de cada componente del sistema, sino que también permite una mayor escalabilidad y facilidad de mantenimiento. Con una arquitectura de este tipo, los servicios pueden evolucionar de manera independiente, lo que garantiza una respuesta rápida a las necesidades cambiantes de los usuarios.

El desarrollo del sistema requiere un enfoque ágil y colaborativo, y la metodología Scrum demostró ser una excelente candidata para gestionar proyectos de software dinámicos. Scrum permite una planificación iterativa e incremental, fomentando la colaboración del equipo y garantizando entregables de alta calidad en plazos ajustados. Esta metodología se ajusta perfectamente a las necesidades del proyecto, donde la flexibilidad y la retroalimentación constante son esenciales.

Para asegurar la calidad del aplicativo desarrollado, es oportuno aplicar estándares internacionales como la norma ISO/IEC 25022. Este estándar proporciona un marco sólido para evaluar la calidad en uso del sistema mediante métricas como Eficacia, Eficiencia y Satisfacción. Estas métricas permiten medir el impacto real del sistema en los usuarios y garantizan que se cumplan los objetivos establecidos.

Finalmente, los estudios relacionados respaldan la viabilidad del proyecto. Trabajos previos que abordan problemas y soluciones similares han demostrado que tecnologías como Flutter y las arquitecturas de microservicios son altamente efectivas para el desarrollo de aplicaciones educativas. Estas experiencias proporcionan un respaldo técnico y metodológico que fortalece los fundamentos del proyecto.

En conjunto, estos elementos establecieron una base sólida para el desarrollo del Portafolio Estudiante, garantizando su alineación con las tendencias tecnológicas y las necesidades institucionales.

II DESARROLLO

En este capítulo se muestra todas las fases del desarrollo utilizando la metodología Scrum, como se muestra en la Fig. 8.

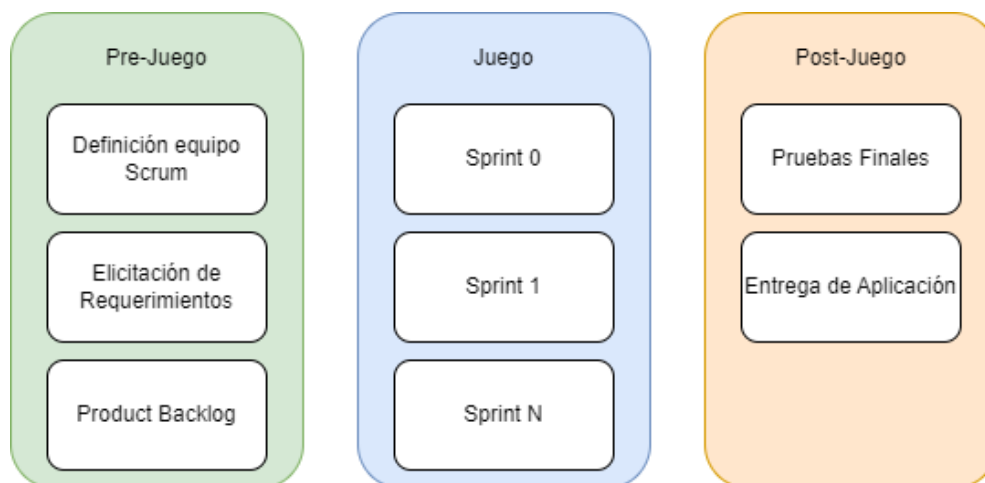


Fig. 8 Diagrama Fases de Desarrollo

2.1 Desarrollo Fase de Pre-Juego

2.1.1 Equipo SCRUM

En la Tabla 2 se indica el equipo Scrum para la implementación del servicio Portafolio Estudiante en la plataforma móvil de este proyecto.

Tabla 2 Roles del Proyecto

Nombre	Rol	Cargo
Antonio Quiña	Product Owner	Director del Trabajo de Grado Docente UTN
Santiago Fierro	Scrum Master Equipo de Desarrollo	Tesista

Dirección de Desarrollo
Tecnológico e Informático

Stakeholder

Departamento TI de la Institución

2.1.2 Elicitación de Requerimientos

Los requisitos se han obtenido a través de historias de usuario como indica la metodología SCRUM. Para esto se realizó una reunión con la Dirección de Desarrollo Tecnológico e Informático de la universidad Técnica del Norte. En esta reunión se procedió a revisar todos los procesos que son oportunos para su implementación en un entorno móvil.

A continuación, se presentan todas las historias de usuarios obtenidas durante este proceso:

Tabla 3 Historia de Usuario 1

Código: HU1	Título: Perfil de Usuario	Prioridad: Baja
Descripción:	Como estudiante necesito ver mi ficha de estudiante, credenciales de Eduroam.	
Criterios de aceptación:	CA1: Al ingresar a este apartado, se debe mostrar la ficha de estudiante que contenga Apellidos y Nombres, Fecha de nacimiento, Dirección, Email, Teléfono, Celular, Lugar de nacimiento, Nacionalidad.	
	CA2: Al ingresar a este apartado, se debe mostrar las siguientes credenciales: Correo institucional, Clave temporal de correo, Usuario Eduroam, Contraseña Eduroam, Información Eduroam, Manual Eduroam.	
	CA3: Al ingresar a este apartado se debe mostrar la foto del estudiante	

Tabla 4 Historia de Usuario 2

Código: HU2	Título: Documentos electrónicos personales	Prioridad: Baja
--------------------	---	------------------------

Descripción:	Como estudiante necesito ver mis documentos personales
Criterios de aceptación:	CA1: Al acceder a esta sección se debe permitir solicitar los certificados de: Asistencia a clases, Segunda y tercera matricula, Matricula, Educación física, Almacén de bodega, Sanción, Récord académico, Biblioteca, Historial académico.
	CA2: Al ingresar se deben mostrar todos los certificados solicitados
	CA3: Al ingresar se deben mostrar los certificados de eventos
	CA4: Al ingresar se deben visualizar los documentos personales como Cédula, Certificado de Bachiller, Ficha Socioeconómica, y Título de Grado si es estudiante de posgrado

Tabla 5 Historia de Usuario 3

Código: HU3	Título: Registro académico	Prioridad: Alta
Descripción:	Como estudiante necesito ver las materias de los periodos activos	
Criterios de aceptación:	CA1: Al listar las materias, si existe más de un periodo académico se debe permitir seleccionar uno para listar sus materias.	
	CA2: Al listar una materia mostrar el Nombre de materia, Docente Asignado, Paralelo, Numero de matrícula, acceso al aula virtual.	

Tabla 6 Historia de Usuario 4

Código: HU4	Título: Calendario de Actividades General	Prioridad: Alta
--------------------	--	------------------------

Descripción:	Como estudiante necesito ver en un calendario de actividades de todas las asignaturas
Criterios de aceptación:	<p>CA1: Al seleccionar un día en el calendario se debe desplegar una ventana con todas las actividades existentes con esa fecha.</p> <p>CA2: Al mostrar las actividades se debe presentar la Descripción de la actividad, Tipo de entrega realizada (Entregado con adjunto, Entregado sin adjunto, Actividad Pendiente), Fecha límite de entrega</p> <p>CA3: Al seleccionar una actividad se debe redireccionar al apartado que permita entregar la tarea.</p>

Tabla 7 Historia de Usuario 5

Código: HU5	Título: Actividades Aula Virtual	Prioridad: Alta
Descripción:	Como estudiante necesito ver un listado de actividades de una única asignatura	
Criterios de aceptación:	<p>CA1: Al mostrar el listado de actividades se debe permitir seleccionar la parcial para filtrar.</p> <p>CA2: Al mostrar el listado actividades se debe presentar la Descripción de la actividad, Tipo de entrega realizada (Entregado con adjunto, Entregado sin adjunto, Actividad Pendiente), Nota, Fecha límite de entrega.</p> <p>CA3: Al seleccionar una actividad se debe mostrar el apartado de detalles y entrega.</p>	

Tabla 8 Historia de Usuario 6

Código: HU6	Título: Detalles de actividad	Prioridad: Alta
--------------------	--------------------------------------	------------------------

Descripción:	Como estudiante necesito ver los detalles de la tarea que me fue asignada
Criterios de aceptación:	<p>CA1: Al ver los detalles se debe mostrar la fecha límite de entrega, la fecha de inicio de actividad, el nombre, la descripción y cualquier recurso cargado por el docente.</p> <p>CA2: Al ver los detalles, si la actividad es de tipo Foro dentro de la plataforma o Evaluación dentro de la plataforma, se debe mostrar un mensaje que indique al estudiante que debe usar la plataforma web</p>

Tabla 9 Historia de Usuario 7

Código: HU7	Título: Entrega de actividad	Prioridad: Alta
Descripción:	Como estudiante necesito entregar la actividad si es posible	
Criterios de aceptación:	<p>CA1: Se debe permitir adjuntar un archivo desde el dispositivo</p> <p>CA2: Al enviar una actividad se debe permitir cargar un archivo de máximo 25 Megabytes y con nombre inferior a 100 caracteres</p> <p>CA3: Al enviar una actividad si el tiempo de entrega se encuentra expirado o no hay intentos disponibles se debe ocultar la opción de envío</p>	

Tabla 10 Historia de Usuario 8

Código: HU8	Título: Recursos Aula Virtual	Prioridad: Media
Descripción:	Como estudiante necesito acceder a los recursos de la materia	

Criterios de aceptación:	CA1: Al mostrar el listado de recursos se debe presentar la Descripción y el tipo de recurso
	CA2: Al acceder al recurso, si este es un documento, audio, video o cualquier otro formato se debe permitir la descarga
	CA3: Al acceder al recurso, si este es un enlace se debe redireccionar al mismo.

Tabla 11 Historia de Usuario 9

Código: HU9	Título: Notas Aula virtual	Prioridad: Media
Descripción:	Como estudiante necesito visualizar mis notas de la asignatura seleccionada	
Criterios de aceptación:	CA1: Al presentar un listado con las actividades se debe mostrar las ponderaciones de cada una para la nota final.	
	CV2: Al presentar el listado se debe permitir seleccionar la parcial para filtrar.	

Tabla 12 Historia de Usuario 10

Código: HU10	Título: Notas	Prioridad: Media
Descripción:	Como estudiante necesito ver mis notas de todas las asignaturas.	
Criterios de aceptación:	CA1: Al listar las notas se debe mostrar la Materia, las Notas de cada parcial y la nota final	

CA2: Al intentar visualizar las notas debo escoger entre Notas Ciclo Actual, Notas por Nivel, Notas por Ciclo Académico

Tabla 13 Historia de Usuario 11

Código: HU11	Título: Horario del Estudiante	Prioridad: Media
---------------------	---------------------------------------	-------------------------

Descripción:	Como estudiante necesito visualizar de una forma intuitiva mi horario de clases
---------------------	---

Criterios de aceptación:	CA1: Al presentar el horario se debe mostrar el Día, Materia, Aula, Hora inicio, Hora fin CA2: Se debe iniciar en el día actual
---------------------------------	--

Tabla 14 Historia de Usuario 12

Código: HU12	Título: Inasistencia del Estudiante	Prioridad: Baja
---------------------	--	------------------------

Descripción:	Como estudiante quiero conocer mis inasistencias a clase.
---------------------	---

Criterios de aceptación:	CA1: Al ingresar a esta sección se debe mostrar el porcentaje de inasistencia CA2: Al ingresar a esta sección se debe mostrar los días que el estudiante asistió a clases y los días que faltó.
---------------------------------	--

Tabla 15 Historia de Usuario 13

Código: HU13	Título: Avance de asignatura	Prioridad: Baja
---------------------	-------------------------------------	------------------------

Descripción:	Como estudiante quiero conocer cómo se desarrolla mi avance en la asignatura.
Criterios de aceptación:	CA1: Al ingresar a esta sección se debe mostrar el porcentaje de avance general de la materia. CA2: Al ingresar a esta sección se debe mostrar la distribución del contenido de la materia.

Una vez recolectado los requisitos se procede a ordenarlos y estimarlos en el Backlog. Esta actividad se hace en conjunto con el Producto Owner, Scrum Master, y el equipo de desarrollo. Se usa una escala que sigue la secuencia de Fibonacci con los valores 1, 2, 3, 5, 8 como se recomienda en el libro *“Agile Estimating and Planning”* [72]. Los resultados obtenidos se muestran en la Tabla 16.

Tabla 16 Product Backlog

Orden	Código	Descripción	Estimación (Puntos de Historia)
1	HU4	Calendario de Actividades General	5
2	HU11	Horario del Estudiante	2
3	HU1	Perfil de Usuario	2
4	HU10	Notas	5
5	HU3	Registro académico	5
6	HU5	Actividades Aula Virtual	5
7	HU9	Notas Aula virtual	5

8	HU8	Recursos Aula Virtual	5
9	HU12	Inasistencia del Estudiante	3
10	HU2	Documentos electrónicos personales	8
11	HU7	Entrega de actividad	8
12	HU6	Detalles de actividad	5
13	HU13	Avance de asignatura	3

2.2 Desarrollo Fase de Juego

2.2.1 Sprint 0

En objetivo de este sprint es establecer la arquitectura que guiará a todo el proyecto, para esto primero se define una arquitectura general que muestra todos los componentes o piezas de software que intervienen en el servicio de Portafolio Estudiante. Después, se establece la estructura que guiará la implementación de la API microservicio, esta será utilizada para futuras implementaciones de nuevos microservicios a excepción de que se considere oportuno cambiarla. Finalmente, se define es esquema arquitectónico para la implementación de la aplicación móvil siguiendo los principios de arquitectura limpia.

- *Definición de Arquitectura General de la Plataforma*

Para la implementación del servicio *Portafolio Estudiante*, se establece una plataforma basada en una arquitectura de microservicios, como se ilustra en la Fig. 9.

La plataforma dispone de un cliente móvil desarrollado en Flutter, cuyas peticiones son gestionadas por un API Gateway encargado de redirigir las solicitudes. Previo a este proceso, el API Gateway centraliza la autenticación, evitando la sobrecarga que supondría comunicar individualmente todos los microservicios con el módulo de seguridad para gestionar la autenticación en cada solicitud.

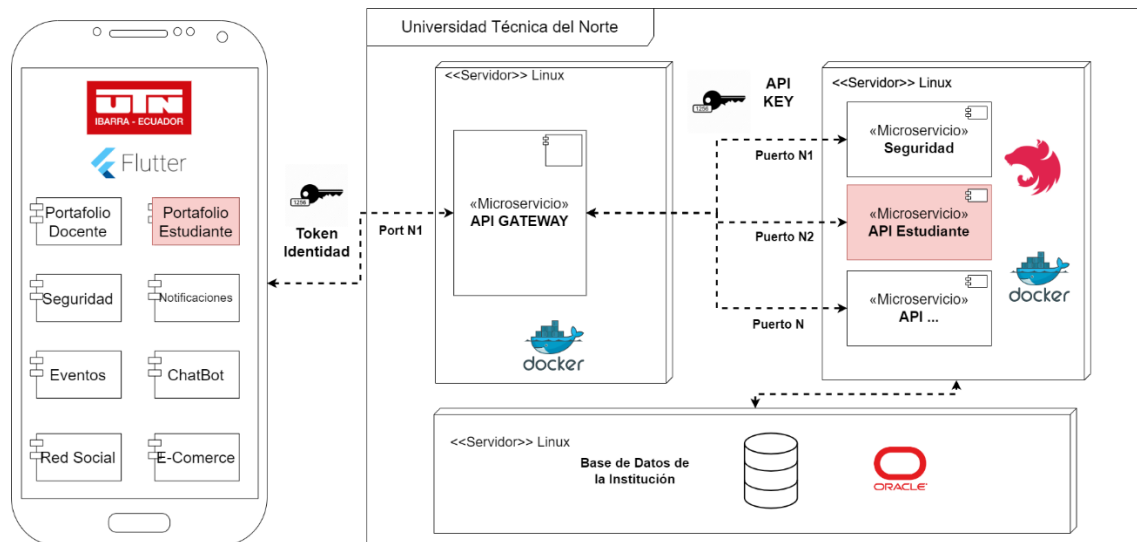


Fig. 9 Arquitectura Plataforma Microservicios

Las solicitudes relacionadas con el Portafolio Estudiante se redirigen a su microservicio específico, desarrollado con el framework NestJS, el cual tiene acceso a la base de datos Oracle de la Universidad Técnica del Norte. Cabe destacar que cada microservicio gestiona la autorización de sus propias funcionalidades, lo que es clave para garantizar la flexibilidad en el desarrollo de cada componente.

- *Definición de la Estructura de la API Microservicio*

El desarrollo del API Microservicio hace uso del framework *NestJS* con *TypeScript* y *express*. La conexión de la base de datos se realiza a través de la librería *TypeORM*, para lo cual es indispensable hacer uso de variables de entorno que aseguren la confidencialidad de las credenciales de acceso. Para mayor facilidad en futuras implementaciones de microservicios se realiza una plantilla alojada en el repositorio GitHub del proyecto, que sigue los principios arquitectónicos que se deberán seguir durante todo el desarrollo. Esta arquitectura se muestra en la Fig. 10.

Los dos principales directorios del proyecto son *src* y *test*. El primero contiene todo el código referente a las funcionalidades del servicio. En *test* se almacenan todas las pruebas de

integración, estas deben estar distribuidos en carpetas por modulo que internamente almacenen pruebas para cada caso de uso.

Dentro del directorio *src* se encuentra el archivo *main.ts* que da inicio al aplicativo. Al mismo nivel se encurtan los directorios *config* y *modules*. El primero es responsable de alojar todas las posibles configuraciones de la aplicación. Por ejemplo, *auth* se encarga de gestionar la autenticación a través filtros y decoradores que permiten validar y extraer el usuario que realiza la petición. Este módulo también se encarga de la autorización exponiendo un decorador con el cual se definen que permisos necesita el usuario para poder ejecutar una función dentro de un *controller*.

El módulo de configuración *exception* se encarga de realizar una gestión de errores centralizada, esto es útil para definir formatos de respuesta que se retornarán a los usuarios con el fin de mitigar el riesgo de filtrar información sensible. Este módulo también gestiona la validación del formato de las peticiones de los usuarios a través de la librería *class-validator*.

El directorio *modules* contiene todos los módulos referentes a los casos de uso del servicio. Cada módulo debe estar dividido en:

- a. *dto*: Contiene las estructuras que serán retornadas al cliente, son útiles para mapear registros de base de datos a través de la librería *class-transformer*.
- b. *entities*: Son las entidades de dominio de la aplicación. Es posible mapear estas entidades utilizando las funcionalidades de la librería *TypeORM*.
- c. *exception*: Contiene los errores específicos de cada módulo, se deben utilizar si se desea personalizar el comportamiento del módulo global de excepciones.
- d. *repository*: Contiene la lógica de acceso a la base de datos y devuelve un *dto* o *entity*. De ser necesario se debe crear un directorio interno llamado *consultas* para alojar exclusivamente constantes que tengan sentencias *SQL*.
- e. *request*: Contiene la estructura del cuerpo de las peticiones como el *body* para peticiones POST o *query* para peticiones GET. Se debe usar la librería *class-validator* para validar el formato.

- f. *nombre.service.ts*: Es la lógica de negocio de la aplicación. A través de la inyección de dependencias accede a los repositorios o a otros *services*. En caso de necesitar más de un *service* es posible crear un directorio.
- g. *nombre.controller.ts*: Se encarga de recibir la peticiones y enviar respuestas. Los servicios son inyectados a través del constructor. En caso de necesitar más de un *controller* es posible crear un directorio.
- h. *nombre.module.ts*: Se encarga de configurar el módulo de NestJS. Aquí se definen los *controller* y *providers*. También es necesario definir la exportación de funcionalidades y la importación de otros módulos.

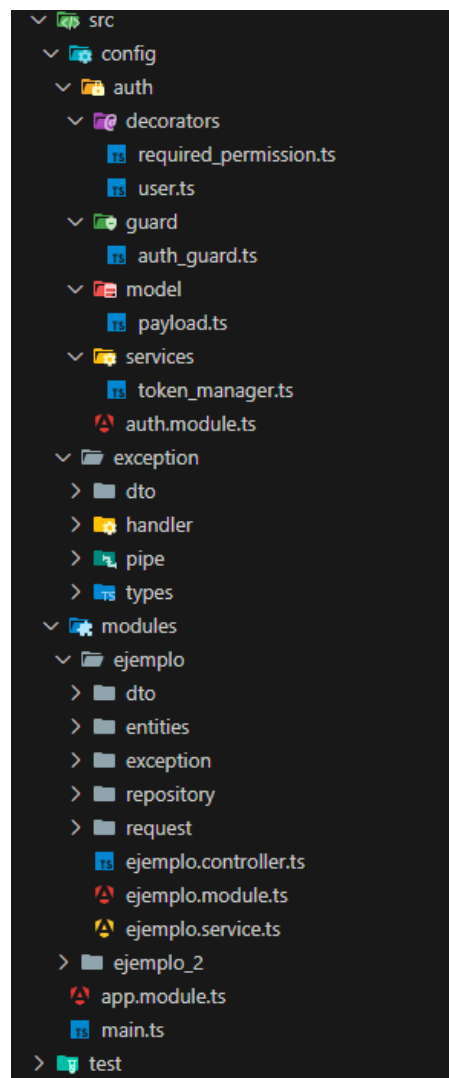


Fig. 10 Estructura Arquitectónica API Microservicio

- *Definición de la Estructura de la APP Móvil*

El desarrollo de la ampliación móvil multiplataforma hace uso del Framework Flutter con el lenguaje de programación Dart. Este sigue las recomendaciones de una arquitectura limpia como se observa en la Fig. 11.

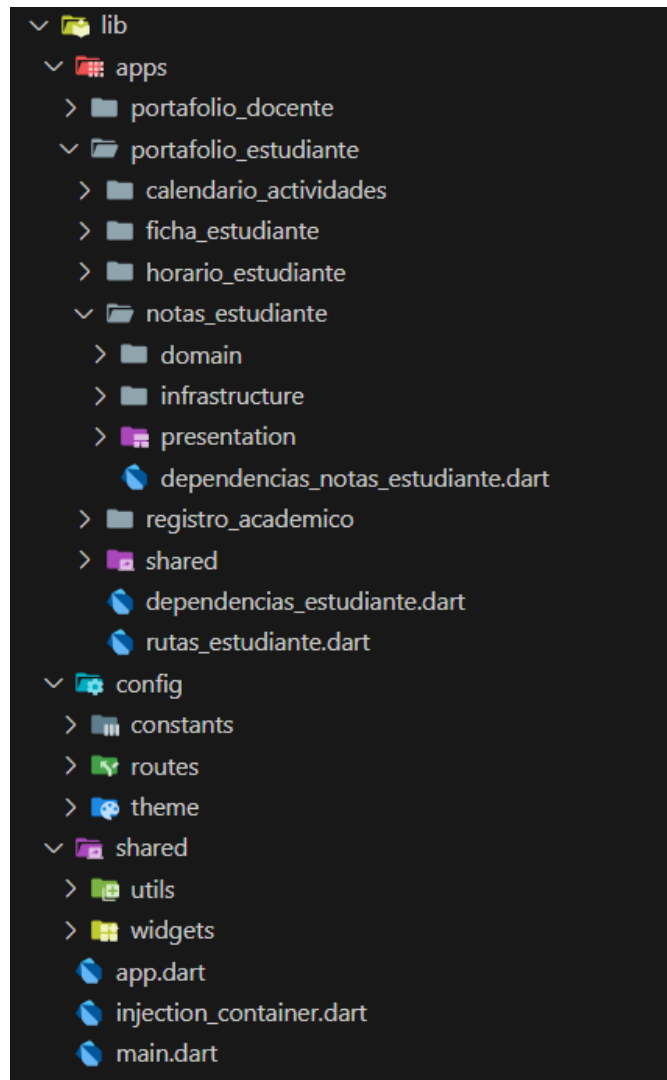


Fig. 11 Estructura Arquitectónica de Aplicación Móvil

Toda la aplicación se encuentra contenida en el directorio `lib` como indica el framework Flutter. Dentro de este directorio se encuentra el archivo `main.dart` que da inicio a la aplicación, aquí se deben inicializar todas las dependencias como las variables de entorno, o librerías para el formato de fechas. Al mismo nivel se encuentra el archivo `injection_container.dart` el cual se puede

observar en la Fig. 12. Este archivo es el encargado de registrar y gestionar las dependencias a través de la librería *get_it*. Es importante mencionar que se registran las dependencias generales de la aplicación y se inicializa el registro de dependencias de cada *sub-aplicación* como *Portafolio Estudiante*, u otras funcionalidades futuras.

```
1 import 'package:dio/dio.dart';
2 import 'package:get_it/get_it.dart';
3 import 'package:utn_movil_app/apps/portafolio_estudiante/dependencias_estudiante.dart';
4 import 'package:utn_movil_app/config/theme/app_theme.dart';
5
6 final sl = GetIt.instance;
7
8 Future<void> initializeDependencies() async {
9   // -----Dependencias Globales-----
10  sl.registerSingleton<Dio>(Dio());
11  sl.registerSingleton<AppTheme>(AppTheme());
12
13  // -----Dependencias Apps-----
14  initDependenciasAppEstudiante(sl);
15 }
16
```

Fig. 12 Registro para Inyección de Dependencias

En el mismo nivel se encuentra el archivo *app.dart* el cual es el *widget* principal que carga las rutas y temas de la aplicación. Estas configuraciones se encuentran en la carpeta *config*. Las rutas son gestionadas por medio de la librería *go_router*. Dentro del archivo principal de enrutamiento únicamente se ejecutan los archivos específicos de rutas de cada *sub-aplicación*. De igual forma dentro de este directorio se encuentran todas las constantes que se utilizan dentro del sistema.

En el directorio *shared* se encuentran dos subcarpetas, La primera carpeta *utils*, es responsable de alojar cualquier funcionalidad que se reutilice a lo largo de toda la aplicación como abrir un enlace web o mostrar una ventana modal emergente. El segundo directorio corresponde a *widgets*, aquí se almacenan todos los componentes visuales reutilizables como la barra de navegación, o un mensaje de error.

Finalmente, la carpeta *apps* contiene todas las *sub-aplicaciones* como el Portafolio Estudiante y el Portafolio Docente. Cada una de estas es responsable de registrar sus rutas internas y sus servicios para inyección de dependencias. La carpeta *shared* al igual que su antecesora contiene

todo el código que puede ser reutilizado dentro de la *sub-aplicación* como la navegación entre pantallas. Al mismo nivel se encurtan todas las carpetas referentes a los casos de uso o módulos de la aplicación. Cada uno de estos directorios están subdivididos en las capas de dominio, infraestructura y presentación, como se observa en la Fig. 13.

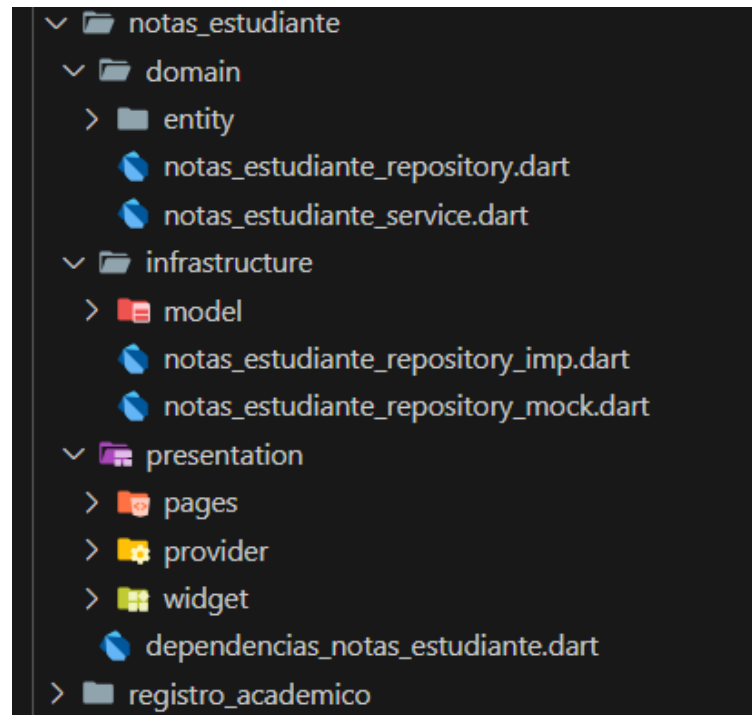


Fig. 13 Arquitectura de cada Caso de Uso en la Aplicación Móvil

La capa de dominio se encarga de la lógica del negocio, esta contiene una carpeta de entidades las cuales son el núcleo de la aplicación. También contiene los servicios que son las funcionalidades que satisfacen los casos de uso. Finalmente se encuentra una interfaz de repositorio, en esta se define los accesos a los datos sin llegar a implementarlos. Esto se realiza siguiendo el principio de inversión de dependencias para asegurar una separación de funcionalidades.

La capa de infraestructura es la encargada de implementar todo lo que se define en la capa de dominio. Las interfaces se conectan con la API Microservicios para obtener datos reales, también es posible implementar *mocks* que devuelvan datos falsos para facilidad de desarrollo y pruebas.

El directorio de modelos contiene toda la lógica para mapear las respuestas de la API hacia las entidades.

Finalmente se encuentra la capa de presentación subdividida en 3. La carpeta *providers* contiene toda la lógica para la gestión de estados de la aplicación, como realizar una petición a la capa de dominio y mientras se espera la respuesta marcar un estado de carga para informar al usuario. La siguiente carpeta es *widget* en esta se almacenan todos los componentes visuales utilizados en las vistas, esto se realiza con el fin de evitar sobrecargar archivos con muchos componentes. La carpeta de *pages* es la encargada de almacenar todas las pantallas del caso de uso. Utilizan los componentes para presentarlos en pantalla, y los *providers* para gestionar estados. Es importante mencionar que toda la lógica visual debe estar en los *providers*, las páginas únicamente deben presentar información.

Es oportuno destacar que existe un archivo para el registro de inyección de dependencias como se muestra en la Fig. 14. Este es ejecutado a través del archivo general de dependencias de la *sub-aplicación*. Aquí se registran todos los componentes creados dentro del caso de uso. Los *providers*, *services* y *repositories* son inyectados para presentar una pantalla. Es necesario evaluar que tipo de registro se va a realizar para cada situación, es común registrar los repositorios como *Lazy Singleton* al igual que los servicios, sin embargo, los *providers* se registran con *Factory*.

```
11
12  initDependenciasNotasEstudiante(GetIt getIt) {
13      // -----Repository-----
14      getIt.registerLazySingleton<NotasEstudianteRepository>(
15          AppConstants.environment == Environment.mock
16              ? () => NotasEstudianteRepositoryMock()
17              : () => NotasEstudianteRepositoryImp(getIt.get());
18
19      // -----Services-----
20      getIt.registerLazySingleton<NotasEstudianteService>(
21          () => NotasEstudianteService(getIt.get());
22
23      // -----Providers-----
24      getIt.registerFactory<NotasEstudianteCicloActualProvider>(
25          () => NotasEstudianteCicloActualProvider(getIt.get());
26
27      getIt.registerFactory<NotasEstudiantePorCicloProvider>(
28          () => NotasEstudiantePorCicloProvider(getIt.get());
29
30      getIt.registerFactory<NotasEstudiantePorNivelProvider>(
31          () => NotasEstudiantePorNivelProvider(getIt.get());
32  }
33
```

Fig. 14 Inyección de Dependencias del Caso de Uso Notas

2.2.2 Sprint 1

- *Planificación - Sprint 1*

La reunión se llevó a cabo con los roles Scrum Master, Producto Owner y Equipo de Desarrollo. El resultado fue la obtención del sprint backlog que se muestra en la Tabla 17. No se planificó 4 horas en caso de imprevistos.

Tabla 17 Sprint Backlog - Sprint 1

Código	Nombre	Tareas	Horas
HU4	Calendario de Actividades General	Implementar endpoint en API	2
		Desarrollar pruebas unitarias en API	1
		Desarrollar funcionalidad en APP	4
HU11	Horario del Estudiante	Implementar endpoint en API	1
		Desarrollar pruebas unitarias en API	1
		Desarrollar funcionalidad en APP	2
HU1	Perfil de Usuario	Implementar endpoint en API	1
		Desarrollar pruebas unitarias en API	1
		Desarrollar funcionalidad en APP	2
HU10	Notas	Desarrollar endpoint notas por nivel	2
		Desarrollar endpoint notas por ciclo	2
		Desarrollar endpoint notas ciclo actual	2

	Desarrollar pruebas unitarias en API notas por ciclo	1
	Desarrollar pruebas unitarias en API notas por nivel	1
	Desarrollar pruebas unitarias en API notas ciclo actual	1
	Desarrollar funcionalidad en APP notas por ciclo	2
	Desarrollar funcionalidad en APP notas por nivel	2
	Desarrollar funcionalidad en APP notas ciclo actual	2
	Planificación	1.5
	Revisión	2
Reuniones Scrum	Retrospectiva	1
	Daily scrum	1.5
	Total	36

- *Incremento – Sprint 1*

El calendario de actividades que se especifica en la

Tabla 6 Historia de Usuario 4 se observa en la Fig. 15. Al presionar en una fecha específica se despliega la lista de actividades únicamente de ese día. Cuando se presiona en una de las actividades se muestra la ventana con los detalles como se muestra en la Fig. 16. Esta es la primera versión de esta vista y está sujeta a cambios.

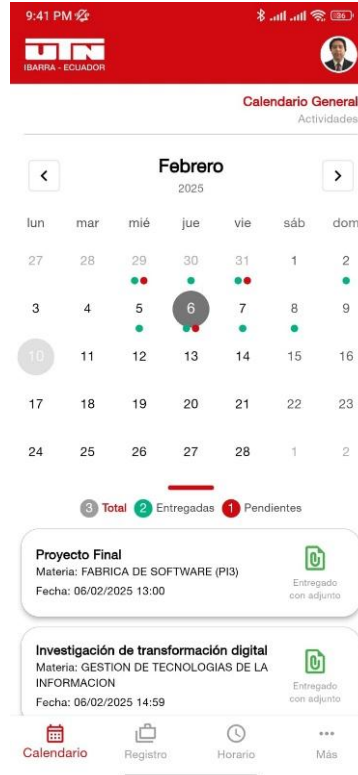


Fig. 15 Vista calendario actividades



Fig. 16 Detalles de actividad (Primera Versión)

En la Fig. 17 se presenta la vista del horario del estudiante, mientras que en la Fig. 18 se presentan los detalles de cada clase. Esto en cumplimiento de Tabla 13.

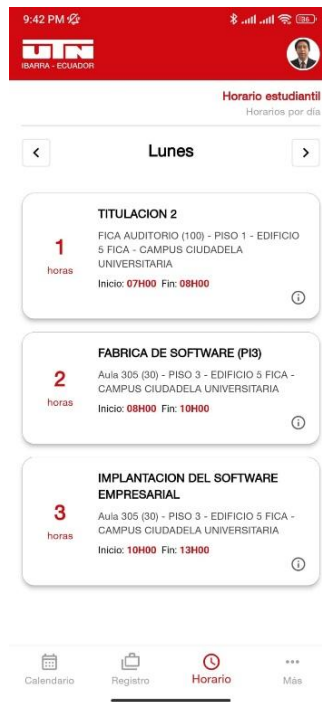


Fig. 17 Vista horario estudiante



Fig. 18 Vista detalles de horario estudiante

Posteriormente en la Fig. 19 se muestran los datos personales del estudiante. En la Fig. 20 se muestran sus credenciales. Esto hace referencia a la Tabla 3.



Fig. 19 Vista perfil estudiante con datos personales

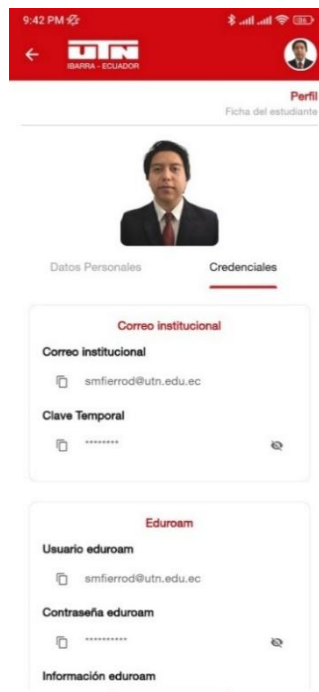


Fig. 20 Vista perfil estudiante con credenciales

Finalmente se desarrolló la Tabla 12. En la Fig. 21 se observa un menú en donde puede escoger que notas el estudiante quiere ver.



Fig. 21 Vista notas estudiante menú

En la Fig. 22 se muestra la vista de las notas del ciclo actual.



Fig. 22 Vista notas estudiante ciclo actual

- *Revisión – Sprint 1*

A continuación, en la Tabla 18 se detalla el cumplimiento de los criterios de aceptación de todas las historias de usuario desarrolladas durante este Sprint.

Tabla 18 Revisión de criterios de aceptación Sprint 1

Historia de Usuario	Criterio de Aceptación	Cumple
H4: Calendario de Actividades General	CA1: Al seleccionar un día en el calendario se debe desplegar una venta con todas las actividades existentes con esa fecha.	SI
	CA2: Al mostrar las actividades se debe presentar la Descripción de la actividad, Tipo de entrega realizada (Entregado con adjunto, Entregado sin adjunto, Actividad Pendiente), Fecha límite de entrega	SI
	CA3: Al seleccionar una actividad se debe redireccionar al apartado que permita entregar la tarea.	SI
H11: Horario del Estudiante	CA1: Al presentar el horario se debe mostrar el Día, Materia, Aula, Hora inicio, Hora fin	SI
	CA2: Se debe iniciar en el día actual	SI
H1: Perfil de Usuario	CA1: Al ingresar a este apartado, se debe mostrar la ficha de estudiante que contenga Apellidos y Nombres, Fecha de nacimiento, Dirección, Email, Teléfono, Celular, Lugar de nacimiento, Nacionalidad.	SI

	CA2: Al ingresar a este apartado, se debe mostrar las siguientes credenciales: Correo institucional, Clave temporal de correo, Usuario Eduroam, Contraseña Eduroam, Información Eduroam, Manual Eduroam.	SI
	CA3: Al ingresar a este apartado se debe mostrar la foto del estudiante	SI
H10: Notas	CA1: Al listar las notas se debe mostrar la Materia, las Notas de cada parcial y la nota final	SI
	CA2: Al intentar visualizar las notas debo escoger entre Notas Ciclo Actual, Notas por Nivel, Notas por Ciclo Académico	SI

- *Retrospectiva – Sprint 1*

En la reunión retrospectiva del sprint, se destacó como acierto la decisión de utilizar los colores principales de la institución, lo cual refuerza la identidad visual del proyecto y asegura coherencia con los lineamientos establecidos. Asimismo, se valoró positivamente la arquitectura implementada en el sprint anterior, ya que ha facilitado la generación de código limpio y mantenible. Por otro lado, como problema recurrente se identificó cierta dificultad en la aplicación de estilos dentro del marco de Flutter, lo cual ha generado demoras y ajustes adicionales. Para mitigar estos inconvenientes, se propuso como mejora la adquisición de un curso básico de Flutter, con el fin de fortalecer las habilidades del equipo y optimizar el flujo de trabajo en los próximos Sprints.

2.2.3 Sprint 2

- *Planificación - Sprint 2*

La reunión se llevó a cabo con los roles Scrum Master, Producto Owner y Equipo de Desarrollo. El resultado fue la obtención del sprint backlog que se muestra en la Tabla 19. No se planificó 4 horas en caso de imprevistos.

Tabla 19 Sprint Backlog - Sprint 2

Código	Nombre	Tareas	Horas
		Implementar endpoint en API	3
HU3	Registro académico	Desarrollar pruebas unitarias en API	1
		Desarrollar funcionalidad en APP	4
		Desarrollar interfaz global para el aula virtual	5
HU5	Actividades Aula Virtual	Implementar endpoint en API	3
		Desarrollar pruebas unitarias en API	1
		Desarrollar funcionalidad en APP	4
		Implementar endpoint en API para la nota parcial	2
HU9	Notas Aula virtual	Desarrollar pruebas unitarias para la nota parcial	1
		Desarrollar funcionalidad en APP de la nota parcial	1
		Implementar endpoint en API para detalles de nota	2

	Desarrollar pruebas unitarias en API para detalles de nota	1
	Desarrollar funcionalidad en APP para detalles de nota	2
Reuniones Scrum	Planificación	1.5
	Revisión	2
	Retrospectiva	1
	Daily scrum	1.5
Total		36

- *Incremento – Sprint 2*

En la Fig. 23 se muestra el listado de aulas virtuales como se indica en Tabla 5.



Fig. 23 Vista registro académico

En la Fig. 24 se observa el listado de actividades que existen dentro de la materia como se indica en la Tabla 7.



Fig. 24 Vista actividades aula virtual

Finalmente se desarrolló la Tabla 11. En la Fig. 25 se observa el listado de notas del aula virtual.



Fig. 25 Vista notas aula virtual

- *Revisión – Sprint 2*

En la Tabla 20 se muestra en detalle la revisión de los criterios de aceptación para este Sprint.

Tabla 20 Revisión de criterios de aceptación Sprint 2

Historia de Usuario	Criterio de Aceptación	Cumple
HU3: Registro académico	CA1: Al listar las materias, si existe más de un periodo académico se debe permitir seleccionar uno para listar sus materias.	SI
	CA2: Al listar una materia mostrar el Nombre de materia, Docente Asignado, Paralelo, Numero de matrícula, acceso al aula virtual.	SI
HU5: Actividades Aula Virtual	CA1: Al mostrar el listado de actividades se debe permitir seleccionar la parcial para filtrar.	SI
	CA2: Al mostrar el listado actividades se debe presentar la Descripción de la actividad, Tipo de entrega realizada (Entregado con adjunto, Entregado sin adjunto, Actividad Pendiente), Nota, Fecha límite de entrega.	SI
	CA3: Al seleccionar una actividad se debe mostrar el apartado de detalles y entrega.	SI
HU9: Notas Aula virtual	CA1: Al presentar un listado con las actividades se debe mostrar las ponderaciones de cada una para la nota final.	SI

CV2: Al presentar el listado se debe permitir seleccionar la parcial para filtrar.

SI

- *Retrospectiva – Sprint 2*

En la reunión se resaltó como acierto haber considerado el uso de tamaños grandes de letra configurados por algunos usuarios en sus dispositivos. Esto permitió garantizar una mejor accesibilidad y experiencia de usuario, alineándose con las buenas prácticas de diseño inclusivo. Además, se confirmó la decisión de adquirir un curso avanzado de Flutter, lo que permitirá al equipo profundizar en conceptos más complejos y resolver eficientemente los desafíos técnicos que surjan en los futuros desarrollos.

2.2.4 Sprint 3

- *Planificación - Sprint 3*

La reunión se llevó a cabo con los roles Scrum Master, Producto Owner y Equipo de Desarrollo. El resultado fue la obtención del sprint backlog que se muestra en la Tabla 21. No se planificó 4 horas en caso de imprevistos.

Tabla 21 Sprint Backlog - Sprint 3

Código	Nombre	Tareas	Horas
		Implementar endpoint en API	2
		Desarrollar pruebas unitarias en API	1
HU8	Recursos Aula Virtual	Desarrollar funcionalidad en APP	4
		Desarrollar procedimiento almacenado en la base de datos para obtener lo documentos del directorio	4

HU12	Inasistencia del Estudiante	Implementar endpoint en API	3
		Desarrollar pruebas unitarias en API	1
		Desarrollar funcionalidad en APP	4
HU2	Documentos electrónicos personales	Implementar endpoint en API	6
		Desarrollar pruebas unitarias en API	2
		Desarrollar funcionalidad en APP	3
Reuniones Scrum	Planificación	1.5	
	Revisión	2	
	Retrospectiva	1	
	Daily scrum	1.5	
Total			36

- *Incremento – Sprint 3*

En la Fig. 26 se observa la vista referente a la Tabla 10.

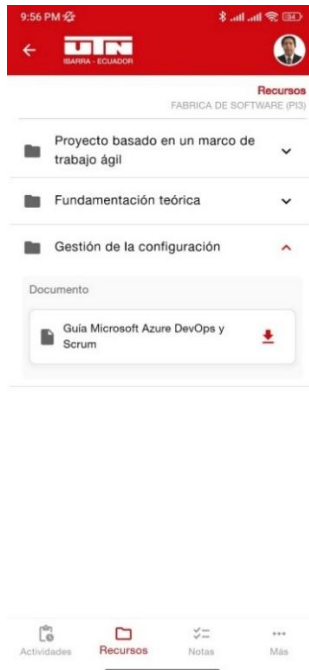


Fig. 26 Vista recursos aula virtual

En la Fig. 27 se observa la vista que se hace referencia en la Tabla 14.



Fig. 27 Vista de asistencia del estudiante

Finalmente, en la Fig. 28 se observa el menú de la sección de documentos del estudiante.



Fig. 28 Vista menú documentos estudiante

En al Fig. 29 se observa la lista de certificados del estudiante de todos los eventos asistidos.



Fig. 29 Vista certificados de eventos

En la Fig. 30 se observan todos los certificados solicitados por el estudiante.



Fig. 30 Vista historial de certificados

En la Fig. 31 se observa el formulario de solicitud de un certificado.

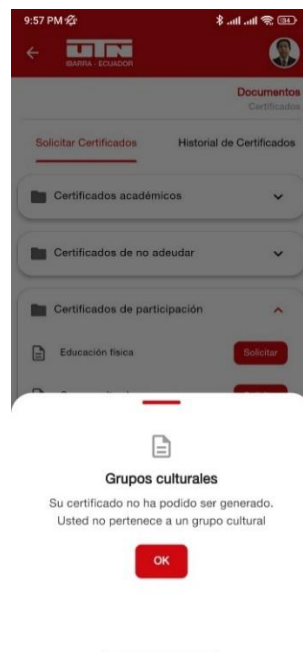


Fig. 31 Vista formulario de solicitud de certificado

- *Revisión – Sprint 3*

En la Tabla 22 se muestra en detalle la revisión de los criterios de aceptación para este Sprint.

Tabla 22 Revisión de criterios de aceptación Sprint 3

Historia de Usuario	Criterio de Aceptación	Cumple
HU8: Recursos Aula Virtual	CA1: Al mostrar el listado de recursos se debe presentar la Descripción y el tipo de recurso	SI
	CA2: Al acceder al recurso, si este es un documento, audio, video o cualquier otro formato se debe permitir la descarga	SI
	CA3: Al acceder al recurso, si este es un enlace se debe redireccionar al mismo.	SI
HU12: Inasistencia del Estudiante	CA1: Al ingresar a esta sección se debe mostrar el porcentaje de inasistencia	SI
	CA2: Al ingresar a esta sección se debe mostrar los días que el estudiante asistió a clases y los días que faltó.	SI
HU2: Documentos electrónicos personales	CA1: Al acceder a esta sección se debe permitir solicitar los certificados de: Asistencia a clases, Segunda y tercera matricula, Matricula, Educación física, Almacén de bodega, Sanción, Récord académico, Biblioteca, Historial académico.	SI

CA2: Al ingresar se deben mostrar todos los certificados solicitados	SI
CA3: Al ingresar se deben mostrar los certificados de eventos	SI
CA4: Al ingresar se deben visualizar los documentos personales como Cédula, Certificado de Bachiller, Ficha Socioeconómica, y Título de Grado si es estudiante de posgrado	SI

- *Retrospectiva – Sprint 3*

Se destacó como un acierto clave haber comenzado a considerar las implicaciones del despliegue, lo que permitió al equipo prever y planificar mejor los posibles desafíos asociados con la publicación y mantenimiento del proyecto en producción. Además, en la sección de documentos personales, se valoró positivamente la decisión de permitir que la aplicación reciba documentos de cualquier tipo, delegando al backend la lógica de agruparlos y gestionarlos. Esta estrategia garantiza flexibilidad y escalabilidad, ya que evita la necesidad de actualizaciones en la aplicación móvil cada vez que se desee incorporar nuevos tipos de documentos, optimizando así el mantenimiento en la fase de producción.

2.2.5 Sprint 4

- *Planificación - Sprint 4*

La reunión se llevó a cabo con los roles Scrum Master, Producto Owner y Equipo de Desarrollo. El resultado fue la obtención del sprint backlog que se muestra en la Tabla 23. No se planificó 4 horas en caso de imprevistos.

Tabla 23 Sprint Backlog - Sprint 4

Código	Nombre	Tareas	Horas
HU7	Entrega de actividad	Implementar endpoint en API	3
		Desarrollar pruebas unitarias en API	2
		Desarrollar funcionalidad en APP	4
		Desarrollar procedimiento almacenado en la base de datos para guardar un archivo adjunto	3
		Desarrollar procedimiento almacenado en la base de datos para entregar una tarea	3
HU6	Detalles de actividad	Implementar endpoint en API	3
		Desarrollar pruebas unitarias en API	1
		Desarrollar funcionalidad en APP	5
HU13	Avance de asignatura	Implementar endpoint en API	2
		Desarrollar pruebas unitarias en API	2
		Desarrollar funcionalidad en APP	2
Reuniones Scrum	Planificación	1.5	
	Revisión	2	
	Retrospectiva	1	
	Daily scrum	1.5	

- *Incremento – Sprint 4*

En la Fig. 32 se observa el porcentaje de avance y distribución de la materia como se indica en la Tabla 15.



Fig. 32 Vista avance de asignatura

En la Fig. 33 se muestran los detalles de una actividad y el formulario para entregar la tarea como se indica en la Tabla 8 y Tabla 9.

The screenshot shows the details for the activity 'Planificación - Sprint2'. It includes submission options (Detalle, Adjunto, Entrega), submission status (2 attempts remaining, 0.5 points), a description field containing 'Entrega de planificación', a URI field, and an attached file 'UTN-MOVI-Despliegue'. A note at the bottom states 'La actividad no acepta respuestas'.

Fig. 33 Vista detalles asignatura versión final

Además, existe el acceso a la descripción completa de la actividad como se indica en la Fig. 34.

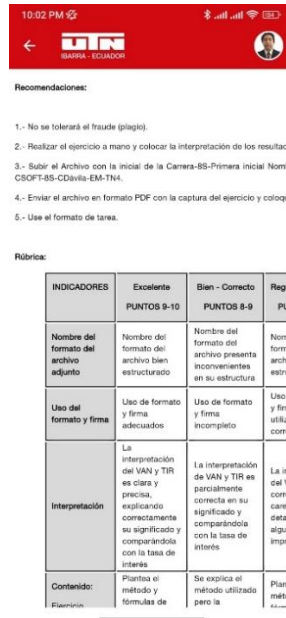


Fig. 34 Vista descripción actividad

Finalmente es posible adjuntar archivos desde el dispositivo, tomar fotos o escanear un documento como se muestra en la Fig. 35.



Fig. 35 Vista opciones de adjuntar archivos

- *Revisión – Sprint 4*

En la Tabla 24 se muestra en detalle la revisión de los criterios de aceptación para este Sprint.

Tabla 24 Revisión de criterios de aceptación Sprint 4

Historia de Usuario	Criterio de Aceptación	Cumple
HU7: Entrega de actividad	CA1: Se debe permitir adjuntar un archivo desde el dispositivo	SI
	CA2: Al enviar una actividad se debe permitir cargar un archivo de máximo 25 Megabytes y con nombre inferior a 100 caracteres	SI
	CA3: Al enviar una actividad si el tiempo de entrega se encuentra expirado o no hay intentos disponibles se debe ocultar la opción de envío	SI
HU6: Detalles de actividad	CA1: Al ver los detalles se debe mostrar la fecha límite de entrega, la fecha de inicio de actividad, el nombre, la descripción y cualquier recurso cargado por el docente.	SI
	CA2: Al ver los detalles, si la actividad es de tipo Foro dentro de la plataforma o Evaluación dentro de la plataforma, se debe mostrar un mensaje que indique al estudiante que debe usar la plataforma web	SI
HU13: Avance de asignatura	CA1: Al ingresar a esta sección se debe mostrar el porcentaje de avance general de la materia.	SI

CA2: Al ingresar a esta sección se debe mostrar
la distribución del contenido de la materia.

SI

- *Retrospectiva – Sprint 4*

Se destacó como un acierto implementar validaciones dinámicas directamente en el servidor, eliminando la necesidad de actualizaciones en la aplicación. Asimismo, se confirmó que todas las funcionalidades incluidas en el alcance del proyecto fueron implementadas con éxito, alcanzando los objetivos planteados y consolidando un progreso significativo en el desarrollo.

2.3 Desarrollo Fase de Post-Juego

Durante esta fase, se llevó a cabo el despliegue de la arquitectura necesaria para poner en funcionamiento el primer prototipo de la aplicación. Para lograrlo, se utilizó Docker como herramienta de contenedores, lo que permitió aislar cada componente del sistema y garantizar un entorno de ejecución homogéneo en diferentes máquinas.

El objetivo principal de este despliegue fue validar el correcto funcionamiento de la aplicación en un entorno controlado, permitiendo realizar pruebas funcionales, de rendimiento y de seguridad. Estas validaciones, que se detallan en el siguiente capítulo, fueron fundamentales para identificar posibles mejoras y garantizar que la aplicación cumpliera con los requisitos establecidos.

Paralelamente, se estableció un plan de transferencia tecnológica con el propósito de facilitar la continuidad del proyecto por parte de futuros estudiantes o desarrolladores interesados. Se generaron archivos Dockerfile y docker-compose.yml para simplificar el despliegue del entorno de desarrollo y producción, reduciendo la complejidad técnica para los nuevos colaboradores. Además, se propuso la implementación de sesiones de capacitación y la utilización de herramientas colaborativas para asegurar una transición fluida del conocimiento y evitar la dependencia de un equipo específico.

III VALIDACIÓN DE RESULTADOS

3.1 Definición de Modelo de Calidad en Uso

En conjunto con el Product Owner y el Scrum Master, se definieron las características de calidad en uso que se usaran según la Norma ISO/IEC 25010 especificada dentro de la familia ISO/IEC 25000 [63]. Las características escogidas son: Eficacia, Eficiencia y Satisfacción. También se definieron sub características y cuál es su peso dependiendo de su impacto en el negocio. Esto se realizó tomando en cuenta la experiencia previa del Product Owner.

Tabla 25 Definición del modelo de calidad en uso

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

3.2 Medición de Calidad en Uso

Para realizar la medición de cada una de las características y sub características se utilizó la norma ISO/IEC 25022 [73]. La eficacia y eficiencia se midió a través de un taller practico y la satisfacción a través de la encuesta SUS.

3.2.1 Definición de Muestra poblacional

La muestra se obtuvo a través de un muestreo no probabilístico por conveniencia debido a la gran cantidad de estudiantes. Se seleccionaron 66 estudiantes de la carrera de Software de la facultad de ciencias aplicadas (FICA) como se muestra en el **Anexo A**. Dentro de este grupo se identificó a 5 usuarios expertos que ya tiene experiencia usando el portal web de la institución y también se encuentran familiarizados con el uso de la aplicación. Dichos estudiantes son aquellos con id 44, 48, 49, 64, 66 que se observan en el **Anexo C** y **Anexo D**.

3.2.2 Taller Práctico

El taller práctico se diseñó con 3 objetivos, cada objetivo tiene 3 tareas para su cumplimiento como se muestra en la Tabla 26. Para realzar el taller el usuario debe tener una cuenta en el sistema universitario y debe iniciar sesión en la aplicación. Todos los objetivos iniciaron desde la pantalla principal del aplicativo, ver **Anexo B**.

Tabla 26 Objetivos y tareas del taller práctico

Nro.	Objetivo	Tarea
1	Enviar una actividad escaneando un documento	Buscar y seleccionar la actividad en el calendario de actividades
		Llenar el campo descripción
		Escanear un documento como adjunto en la opción Escanear documento y entregar tarea

		Seleccionar la materia en el registro académico
2	Ver la nota de trabajos de la primera parcial de una materia	Ingresar al apartado de notas
		Seleccionar la primera parcial y expandir apartado de trabajos
3	Ver las horas asistidas a clases en un día específico de una materia	Seleccionar la materia en el registro académico
		Ingresar al apartado más y luego asistencia
		Seleccionar el día del mes de diciembre
4	Solicitar certificado de matricula	Ingresar al menú más y después documentos
		Seleccionar y completa la solicitud del certificado deseado
		Revisar el estado de la solicitud en el apartado Historial de Certificados
5	Descargar un recurso de una materia	Seleccionar la materia en el Registro Académico
		Ingresar al apartado de Recursos
		Desplegar la primera carpeta y descargar un recurso
6	Ver los certificados de evento generados en el año 2024	Seleccionar el menú Más y luego Documentos
		Seleccionar la opción Eventos
		Definir el rango de búsqueda de Enero 2024 a Diciembre 2024

		Buscar y seleccionar la actividad en el Calendario de Actividades
7	Modificar entrega de tarea (Segundo intento)	Cambiar el campo descripción
		Eliminar el archivo adjunto existente
		Tomar una fotografía usando la opción Tomar foto y entregar tarea

3.2.3 Encuesta SUS

La encuesta SUS (System Usability Scale) propone 10 preguntas para medir el nivel de usabilidad de un sistema como se muestra en la Tabla 27. Se hace uso de la escala de Likert con 5 puntos con los valores de 1 que significa Totalmente en desacuerdo y 5 que significa Totalmente de acuerdo [74], [75].

Tabla 27 Preguntas encuesta SUS [75]

Código	Pregunta
P1	¿Considera usted qué usaría este sistema frecuentemente?
P2	¿Considera usted qué este sistema es innecesariamente complejo?
P3	¿Considera usted qué el sistema fue fácil de usar?
P4	¿Considera usted qué necesitaría ayuda de una persona con conocimientos técnicos para usar este sistema?
P5	¿Considera usted qué las funciones de este sistema están bien integradas?

P6	¿Considera usted que el sistema es inconsistente?
P7	¿Considera usted que la mayoría de la gente aprendería a usar este sistema en forma rápida?
P8	¿Considera usted que el sistema es difícil de usar?
P9	¿Se siente confiado al usar el sistema?
P10	¿Considera usted que necesita aprender muchas cosas tecnológicas antes de usar el sistema?

Las preguntas P1, P6 y P9 fueron seleccionadas para medir la utilidad y las preguntas P2, P3, P4, P5, P7, P8 y P10 fueron seleccionadas para medir la comodidad.

3.3 Validaciones estadísticas de los instrumentos de medición

3.3.1 Fiabilidad del taller práctico

Para evaluar la fiabilidad del taller práctico, se aplicó la prueba de Kolmogorov-Smirnov a los datos tabulados en el **Anexo C**. Esta prueba permite determinar si los datos siguen una distribución normal, si el valor obtenido es superior a 0.5, se considera que la distribución es normal.

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Errores	,536	66	,000	,292	66	,000
TareasLogrados	,539	66	,000	,164	66	,000
ObjetivosLogrados	,539	66	,000	,164	66	,000

Fig. 36 Test de normalidad del taller práctico

Como se observa en la Fig. 36 la cantidad de errores (0.536), las tareas logradas (0.539) y los objetivos logrados (0.539) si tienen una distribución normal y se puede continuar con su respectivo análisis.

3.3.2 Fiabilidad de la encuesta SUS

Una vez tabulados los datos en el **Anexo D** se procedió a realizar el análisis de factibilidad de la encuesta SUS, para esto se utilizó el método estadístico coeficiente Alfa Cronbach. El coeficiente es un valor entre 0 y 1, donde los valores mayores 0.9 son excelentes, entre 0.8 y 0.9 son buenos, mayores a 0.7 aceptables y menores a 0.6 poseen poca fiabilidad.

El análisis se realizó a las 10 preguntas de la encuesta SUS. Es oportuno mencionar que las preguntas pares de esta encuesta son inversas a las impares, por lo tanto, se realizó una recodificación de los resultados para que todos los valores estén en el mismo sentido.

Reliability Statistics

Cronbach's Alpha	N of Items
,766	10

Fig. 37 Alfa de Cronbach encuesta SUS

Como se observa en la Fig. 37 el coeficiente obtenido (0.766) está dentro de los límites aceptables de fiabilidad, por lo tanto, se procede con el análisis de estos resultados.

3.4 Evaluación del Modelo de Calidad en Uso

Utilizando los datos tabulados en el **Anexo C** y **Anexo D** se procede a detallar los resultados obtenidos de las métricas establecidas en la Tabla 25.

3.4.1 Característica: Eficacia

- *Sub característica: Tareas Completadas*

Para calcular esta métrica se hace uso de la siguiente formula:

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

Donde:

- A = número de tareas completadas
- B = número de tareas

Remplazando los valores A (1444) tareas totales completadas por todos los usuarios y B (1452) tareas totales se obtiene como resultado (0.994) lo que significa que el 99% de las tareas fueron completadas con éxito.

- *Sub característica: Objetivos Logrados*

Para calcular esta métrica se hace uso de la siguiente formula:

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

Donde:

- A = número de objetivos completadas
- B = número de objetivos

Se establecieron 7 objetivos, si una tarea no es completada con éxito, el objetivo que engloba dicha tarea no lo considera como completado. Remplazando los valores A (458) y B (462) se obtiene como resultado (0.991) lo que significa que el 99% de los objetivos fueron alcanzados.

- *Sub característica: Errores en una tarea*

Para calcular esta métrica se hace uso de la siguiente formula:

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

Donde:

- A = número de errores en tareas completadas
- B = número de tareas

Remplazando los valores A (5) y B (1444) se obtiene como resultado (0.996) lo que significa que el 99% de tareas se completaron sin errores.

1.1.2 Característica: Eficiencia

- *Sub característica: Tiempo de Tareas*

Para calcular esta métrica se hace uso de la siguiente formula:

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

Donde:

- A = tiempo empleado por usuarios expertos para completar una tarea
- B = tiempo empleado por usuarios nuevos al completar una tarea

Primero es necesario calcular el tiempo promedio por tarea para los dos grupos de usuarios, las tareas no completadas no se toman en cuenta. Posteriormente se aplica la formula por cada tarea y se calcula el promedio de estos resultados. Finalmente, se obtiene como resultado (0.604) lo que significa que existe una eficiencia del 60% en los tiempos de tareas.

- *Sub característica: Eficiencia del Tiempo*

Para calcular esta métrica se hace uso de la siguiente formula:

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

Donde:

- A = tiempo empleado por usuarios expertos para completar un objetivo

- B = tiempo empleado por usuarios nuevos al completar un objetivo

Primero es necesario calcular el tiempo que tarda cada usuario en completar un objetivo, los objetivos no cumplidos no se toman en cuenta. Posteriormente se calcula el promedio del tiempo por cada objetivo de los dos grupos de usuarios. Se aplica la formula a cada objetivo y se calcula el promedio. Finalmente, se obtiene como resultado (0.619) lo que significa que existe una eficiencia del 61% en los tiempos de los objetivos.

1.1.3 Característica: Satisfacción

- *Sub característica: Utilidad*

Se estableció un peso para cada una de las respuestas de la escala de Likert como se muestra en la Tabla 28.

Tabla 28 Peso respuestas 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

También es importante mencionar que se invirtió el sentido de la pregunta P6 para que todas tengan el mismo sentido.

Tabla 29 Resultado SUS - Utilidad

Pregunta	1	2	3	4	5	Resultado
-----------------	----------	----------	----------	----------	----------	------------------

P1	1	0	0	15	50	62.2 de 66
P6	1	4	6	22	33	56 de 66
P9	0	2	7	17	40	58.6 de 66
Promedio						58.93 de 66

Para calcular esta métrica se hace uso de la siguiente formula:

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

Donde:

- A = usuarios que perciben útil el aplicativo
- B = usuarios encuestados

Reemplazando los valores A (58.93) y B (66) se obtiene el resultado (0.892) lo que significa que el 89% de los usuarios perciben que el aplicativo es útil.

- *Sub característica: Confianza*

Para calcular esta métrica se hace uso de la siguiente formula:

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

Donde:

- A = usuarios con reclamos
- B = usuarios encuestados

Reemplazando los valores A (21) y B (66) se obtiene el resultado (0.681) lo que significa que el nivel de confianza del usuario es 68%.

- *Sub característica: Comodidad*

Se establecieron los pesos que se observan en la Tabla 28 y se invirtió el sentido de las preguntas P2, P4, P8, P10.

Tabla 30 Resultados SUS - Comodidad

Pregunta	1	2	3	4	5	Resultado
P2	6	7	11	22	20	48.2 de 66
P3	0	0	4	26	36	59.4 de 66
P4	4	8	6	12	36	53.2 de 66
P5	0	0	1	24	41	60.8 de 66
P7	0	0	2	23	41	60.6 de 66
P8	6	6	6	13	35	52.6 de 66
P10	4	4	5	14	39	55.6 de 66
Promedio						55.77 de 66

Para calcular esta métrica se hace uso de la siguiente formula:

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

Donde:

- A = usuarios que perciben el aplicativo cómodo
- B = usuarios encuestados

Reemplazando los valores A (55.77) y B (66) se obtiene el resultado (0.845) lo que significa que el 84% de los usuarios perciben que el aplicativo es cómodo.

3.5 Resultados del Modelo de Calidad en Uso

En la Tabla 31 se puede observar el resultado de cada una de las métricas y submétricas con las que se midió la aplicación móvil según Tabla 25. El resultado final es **82.39%** de calidad en uso. La escala presentada en la Fig. 7 indica que el aplicativo móvil se encuentra dentro del **rango satisfactorio** como se muestra en la Fig. 38.



Fig. 38 Rango resultado de Calidad en Uso

Tabla 31 Resultados Calidad en Uso

Característica	Sub Característica	Peso de característica	Peso de Sub Característica	Medición	Resultado	Resultado Característica
Eficacia	Tareas completas	42%	16%	0.994	15.9%	41.71%
	Objetivos logrados		16%	0.991	15.85%	
	Tareas sin errores		10%	0.996	9.96%	
Eficiencia	Tiempo de tareas	32%	16%	0.604	9.66%	19.56%
	Eficiencia del tiempo		16%	0.619	9.9%	
Satisfacción	Utilidad	26%	10%	0.892	8.92%	21.12%
	Confianza		8%	0.681	5.44%	
	Comodidad		8%	0.845	6.76%	
Total						82.39%

CONCLUSIONES

La elaboración de un marco conceptual fue esencial para definir las tecnologías y metodologías más adecuadas para el desarrollo de la aplicación móvil. Este análisis permitió identificar las necesidades de modernización del Portafolio Estudiante de la Universidad Técnica del Norte y su alineación con las tendencias tecnológicas actuales.

La implementación de una arquitectura limpia dividida en 3 capas permitió que el desarrollo de la aplicación móvil fuera modular y desacoplado, lo cual facilita la incorporación de nuevas funcionalidades que se adapte a las necesidades futuras de la institución sin comprometer el funcionamiento general. Para esto se consideró necesario el uso de patrones de diseño como la inyección de dependencias y la inversión de dependencias.

Durante el despliegue se decidió alojar inicialmente todos los microservicios en un único servidor para optimizar los recursos disponibles y simplificar la gestión inicial. A medida que la demanda del sistema aumente, será posible escalar progresivamente, migrando los microservicios a entornos independientes de manera controlada y sin afectar la operatividad del sistema.

La evaluación de la calidad en uso de la aplicación móvil para el servicio del Portafolio Estudiante, basada en la norma ISO/IEC 25022, permitió determinar su desempeño en términos de eficacia, eficiencia y satisfacción. Los resultados obtenidos reflejan un 82.39% de calidad en uso, ubicando a la aplicación dentro del rango satisfactorio según la escala establecida.

RECOMENDACIONES

Se recomienda realizar actualizaciones periódicas de la aplicación móvil del "Portafolio Estudiante" para adaptarse a nuevas necesidades académicas, cambios tecnológicos y retroalimentación de los usuarios. Además, es importante considerar la integración con otros sistemas académicos de la universidad para mejorar la interoperabilidad y la experiencia del usuario.

A medida que la demanda del sistema aumente, se recomienda implementar una estrategia de escalabilidad progresiva que permita migrar los microservicios a entornos separados, optimizando el rendimiento y la disponibilidad del sistema. Para ello, se sugiere el uso de orquestadores como Kubernetes, que faciliten la administración de contenedores Docker y la distribución eficiente de los recursos.

Para garantizar el mantenimiento y la evolución del sistema, es fundamental capacitar al equipo técnico en el uso de la arquitectura implementada, los patrones de diseño utilizados y la gestión de contenedores con Docker.

Se recomienda realizar estudios periódicos sobre la experiencia de los estudiantes con la aplicación móvil para identificar áreas de mejora y garantizar su adopción efectiva. Encuestas, grupos focales y análisis de métricas de uso pueden proporcionar información clave para futuras actualizaciones y mejoras del sistema.

REFERENCIAS

- [1] S. Fu, K. Xue, M. Yang, y X. Wang, “An exploratory study on users’ resistance to mobile app updates: Using netnography and fsQCA”, *Technol Forecast Soc Change*, vol. 191, p. 122479, jun. 2023, doi: 10.1016/J.TECHFORE.2023.122479.
- [2] INEC, “Tecnologías de la información y comunicación”, 2023, Consultado: el 2 de mayo de 2024. [En línea]. Disponible en: https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2023/202307_Tecnologia_de_la_Informacion_y_Comunicacion-TICs.pdf
- [3] C. Bröhl, P. Rasche, J. Jablonski, S. Theis, M. Wille, y A. Mertens, “Desktop PC, tablet PC, or smartphone? an analysis of use preferences in daily activities for different technology generations of a worldwide sample”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10926 LNCS, pp. 3–20, 2018, doi: 10.1007/978-3-319-92034-4_1/TABLES/3.
- [4] 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”, Universidad Tecnica del Norte, Ibarra, 2024. Consultado: el 18 de abril de 2024. [En línea]. Disponible en: <https://repositorio.utn.edu.ec/handle/123456789/15812>
- [5] “ISO/IEC 25022:2016 - Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Measurement of quality in use”. Consultado: el 18 de abril de 2024. [En línea]. Disponible en: <https://www.iso.org/standard/35746.html>
- [6] “ISO/IEC 25010:2023 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model”. Consultado: el 22 de abril de 2024. [En línea]. Disponible en: <https://www.iso.org/standard/78176.html>

- [7] E. Alfredo y B. Bastidas, “Desarrollo de un sistema web para la automatización del proceso de mapeo sistemático de la literatura, y validado mediante un marco de trabajo de calidad de uso basado en las normas ISO/IEC 25000 para mejorar el proceso de investigación en los docentes de la Universidad Técnica del Norte”, feb. 2020, Consultado: el 23 de abril de 2024. [En línea]. Disponible en: <https://repositorio.utn.edu.ec/handle/123456789/10254>
- [8] Universidad Técnica del Norte, “Plan Estratégico de Desarrollo Institucional (PEDI) 2021 - 2025”. Consultado: el 2 de diciembre de 2024. [En línea]. Disponible en: <https://transparencia.utn.edu.ec/wp-content/uploads/2024/01/PEDI-2021-2025.pdf>
- [9] “Objetivos y metas de desarrollo sostenible - Desarrollo Sostenible”. Consultado: el 23 de abril de 2024. [En línea]. Disponible en: <https://www.un.org/sustainabledevelopment/es/sustainable-development-goals/>
- [10] R. Malhotra, D. Kumar, y D. P. Gupta, “An Android Application for Campus Information System”, *Procedia Comput Sci*, vol. 172, pp. 863–868, ene. 2020, doi: 10.1016/J.PROCS.2020.05.124.
- [11] I. A. Lescano Vásquez, “Desarrollo de una aplicación móvil utilizando el framework Flutter para fomentar el área turística del GAD de Pedro Moncayo”, Universidad Técnica del Norte, Ibarra, 2022. Consultado: el 10 de marzo de 2024. [En línea]. Disponible en: <http://repositorio.utn.edu.ec/handle/123456789/13228>
- [12] J. D. Flores Landeta, “Estudio de una arquitectura de microservicios mediante Spring Cloud para el desarrollo del módulo de registro y seguimiento médico de los deportistas en la Federación deportiva de Imbabura”, jul. 2020, Consultado: el 10 de marzo de 2024. [En línea]. Disponible en: <http://repositorio.utn.edu.ec/handle/123456789/10445>
- [13] H. Ünlü, D. E. Kennouche, G. K. Soylu, y O. Demirörs, “Microservice-based projects in agile world: A structured interview”, *Volume 165*, vol. 165, ene. 107334, doi: 10.1016/j.infsof.2023.107334.

- [14] Universidad Técnica del Norte, “Reglamento de Portafolio Universitario”, 2014, Consultado: el 7 de junio de 2024. [En línea]. Disponible en: https://legislacion.utn.edu.ec/wp-content/uploads/2017/12/2.REGLAMENTO_PORTAFOLIO_UNIVERSITARIO-1.pdf
- [15] H. Nindito, A. V. D. Sano, y A. R. Condrobimo, “Comparative study of storing unstructured data type between BasicFile and SecureFile in Oracle Database 12c”, *Proceedings of 2016 International Conference on Information Management and Technology, ICIMTech 2016*, pp. 146–149, may 2017, doi: 10.1109/ICIMTECH.2016.7930319.
- [16] F. Almeida, P. Silva, y F. Araújo, “Performance analysis and optimization techniques for Oracle Relational Databases”, *Cybernetics and Information Technologies*, vol. 19, núm. 2, pp. 117–132, jun. 2019, doi: 10.2478/cait-2019-0019.
- [17] M. Kvet, “Rapid Application Development and data management using Oracle APEX and SQL”, *2024 IEEE 22nd World Symposium on Applied Machine Intelligence and Informatics, SAMI 2024 - Proceedings*, pp. 297–302, 2024, doi: 10.1109/SAMI60510.2024.10432914.
- [18] I. Pastierik y M. Kvet, “Oracle Application Express as a Tool for Teaching Web Software Development”, *2023 Communication and Information Technologies, KIT 2023 - 12th International Scientific Conference, Proceedings, 2023*, doi: 10.1109/KIT59097.2023.10297067.
- [19] Universidad Central del Ecuador, “Log in to the site | UCE”. Consultado: el 24 de octubre de 2024. [En línea]. Disponible en: <https://uvirtual.uce.edu.ec/login/index.php>
- [20] Marzuki, A. N. Wulyani, Hidayati, M. R. I. M. Sata, y D. Rusdin, “Overcoming challenges: Indonesian EFL teachers’ strategies for using moodle in high schools”, *Social Sciences & Humanities Open*, vol. 10, p. 101175, ene. 2024, doi: 10.1016/J.SSAHO.2024.101175.
- [21] Universidad Espiritu Santo, “Blackboard Learn UEES”. Consultado: el 24 de octubre de 2024. [En línea]. Disponible en: <https://uees.blackboard.com/>

- [22] M. H. Albahiri y A. A. M. Alhaj, “The Impact of Utilizing YouTube Videos via the Blackboard Platform on Developing the Interpreting Skills of Saudi Translation Students: King Khalid University Faculty’s Perspectives”, *Journal of Language Teaching and Research*, vol. 15, núm. 2, pp. 598–606, mar. 2024, doi: 10.17507/jltr.1502.28.
- [23] V. Jain y V. Viswanathan, “The Usage and Applications of Mobile Apps”, *Encyclopedia of Mobile Phone Behavior*, pp. 1242–1255, jun. 2015, doi: 10.4018/978-1-4666-8239-9.CH100.
- [24] E. Alam, “Social Perspective on Cell Phone Use in Bangladesh from 2000 to 2023 and future 2023 to 2030”, *SSRN Electronic Journal*, sep. 2023, doi: 10.2139/SSRN.4567525.
- [25] D. D. Felisoni y A. S. Godoi, “Cell phone usage and academic performance: An experiment”, *Comput Educ*, vol. 117, pp. 175–187, feb. 2018, doi: 10.1016/J.COMPEDU.2017.10.006.
- [26] INEC, “Tecnologías de la información y comunicación”, 2023, [En línea]. Disponible en: https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2023/202307_Tecnologia_de_la_Informacion_y_Comunicacion-TICs.pdf
- [27] Universidad de las Américas, “UDLA+ - Apps on Google Play”. Consultado: el 24 de octubre de 2024. [En línea]. Disponible en: <https://play.google.com/store/apps/details?id=com.antonio.udlaapp&hl=en-US>
- [28] Universidad del Azuay, “UDA - Apps on Google Play”. Consultado: el 24 de octubre de 2024. [En línea]. Disponible en: <https://play.google.com/store/apps/details?id=com.almendra.uda&hl=en-US>
- [29] Pontificia Universidad Católica del Ecuador, “PUCE APP - Apps en Google Play”. Consultado: el 24 de octubre de 2024. [En línea]. Disponible en: <https://play.google.com/store/apps/details?id=ec.edu.puce.dipuapp&hl=es-EC>

- [30] Universidad Espiritu Santo, "UEES - Apps en Google Play". Consultado: el 24 de octubre de 2024. [En línea]. Disponible en: <https://play.google.com/store/apps/details?id=com.wnapp.id1680548691455&hl=es-EC>
- [31] Kotlin, "What is cross-platform mobile development?", 2024, Consultado: el 8 de junio de 2024. [En línea]. Disponible en: <https://www.jetbrains.com/help/kotlin-multiplatform-dev/cross-platform-mobile-development.html>
- [32] K. Kishore, S. Khare, V. Uniyal, y S. Verma, "Performance and stability Comparison of React and Flutter: Cross-platform Application Development", *International Conference on Cyber Resilience, ICCR 2022*, 2022, doi: 10.1109/ICCR56254.2022.9996039.
- [33] A. M. Hassan, "JAVA and DART programming languages: Conceptual comparison", *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, núm. 2, pp. 845–849, ene. 2019, doi: 10.11591/ijeecs.v17.i2.pp845-849.
- [34] S. T. Aung, N. Funabiki, L. H. Aung, S. A. Kinari, M. Mentari, y K. H. Wai, "A Study of Learning Environment for Initiating Flutter App Development Using Docker", *Information (Switzerland)*, vol. 15, núm. 4, p. 191, abr. 2024, doi: 10.3390/info15040191.
- [35] P. V. Isha y H. Vishalakshmi Prabhu, "An Approach to Clean Architecture for Microservices Using Python", *7th IEEE International Conference on Computational Systems and Information Technology for Sustainable Solutions, CSITSS 2023 - Proceedings*, 2023, doi: 10.1109/CSITSS60515.2023.10334229.
- [36] R. Martin, "The Clean Architecture", 2012, [En línea]. Disponible en: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>
- [37] S. Boukhary y E. Colmenares, "A clean approach to flutter development through the flutter clean architecture package", *Proceedings - 6th Annual Conference on Computational Science and Computational Intelligence, CSCI 2019*, pp. 1115–1120, dic. 2019, doi: 10.1109/CSCI49370.2019.00211.

- [38] Dart API docs, “get_it”, 2024, Consultado: el 8 de junio de 2024. [En línea]. Disponible en: https://pub.dev/documentation/get_it/latest/
- [39] Y. N. Nugroho, D. S. Kusumo, y M. J. Alibasa, “Clean Architecture Implementation Impacts on Maintainability Aspect for Backend System Code Base”, *2022 10th International Conference on Information and Communication Technology, ICOICT 2022*, pp. 134–139, 2022, doi: 10.1109/ICOICT55009.2022.9914890.
- [40] Flutter.cn, “Dio 5.4.3+1 | Dart package”, 2024, Consultado: el 13 de junio de 2024. [En línea]. Disponible en: <https://pub.dev/packages/dio>
- [41] Pinch.nl, “Floor 1.5.0 | Flutter package”, 2024, Consultado: el 13 de junio de 2024. [En línea]. Disponible en: <https://pub.dev/packages/floor>
- [42] E. Windmill, “Flutter in Action”, pp. 2–368, 2019, Consultado: el 13 de junio de 2024. [En línea]. Disponible en: <https://ieeexplore.ieee.org/document/10280403>
- [43] B. De, “Introduction to APIs”, *API Management*, pp. 1–14, 2017, doi: 10.1007/978-1-4842-1305-6_1.
- [44] A. Lercher, J. Glock, C. Macho, y M. Pinzger, “Microservice API Evolution in Practice: A Study on Strategies and Challenges”, *Journal of Systems and Software*, vol. 215, p. 112110, sep. 2024, doi: 10.1016/J.JSS.2024.112110.
- [45] “2024 Stack Overflow Developer Survey”. Consultado: el 3 de diciembre de 2024. [En línea]. Disponible en: <https://survey.stackoverflow.co/2024/>
- [46] “Top Programming Languages 2024 - IEEE Spectrum”. Consultado: el 3 de diciembre de 2024. [En línea]. Disponible en: <https://spectrum.ieee.org/top-programming-languages-2024>
- [47] J. Bogner y M. Merkel, “To Type or Not to Type? A Systematic Comparison of the Software Quality of JavaScript and TypeScript Applications on GitHub”, *Proceedings - 2022 Mining*

- Software Repositories Conference, MSR 2022*, pp. 658–669, 2022, doi: 10.1145/3524842.3528454.
- [48] D. Achiro, R. Alowo, y G. Nkhonjera, “Implementing a Groundwater Monitoring System in the Jukskei River Catchment: a TypeScript and MySQL Approach”, *International Conference on Electrical, Computer and Energy Technologies, ICECET 2023*, p. Town, doi: 10.1109/ICECET58911.2023.10389560.
- [49] T. Nguyen, “Is TypeScript Faster Than JavaScript? A Comprehensive Comparison - Frontend Mag”, 2022, Consultado: el 9 de julio de 2024. [En línea]. Disponible en: <https://www.frontendmag.com/insights/is-typescript-faster-than-javascript/>
- [50] NestJS, “Documentation | NestJS - A progressive Node.js framework”, 2017, Consultado: el 9 de julio de 2024. [En línea]. Disponible en: <https://docs.nestjs.com/>
- [51] M. P. Prasad y U. Padma, “Performance Analysis of ExpressJS and Fastify in NestJS”, *Lecture Notes in Electrical Engineering*, vol. 1066 LNEE, pp. 1037–1049, 2023, doi: 10.1007/978-981-99-4634-1_82.
- [52] A. Duc Pham Abstract Author Pham y A. Duc, “DEVELOPING BACK-END OF A WEB APPLICATION WITH NESTJS FRAMEWORK Title of publication Developing back-end of a web application with NestJS framework Case: Integrify Oy’s student management system”, 2020, Consultado: el 9 de julio de 2024. [En línea]. Disponible en: https://www.theseus.fi/bitstream/handle/10024/353200/Pham_Duc.pdf?sequence=2&isAllowed=y
- [53] S. Van Deursen y M. Seemann, “Dependency Injection: Principles, Practices, Patterns”, p. 507, 2019, Consultado: el 9 de julio de 2024. [En línea]. Disponible en: <https://ieeexplore.ieee.org/document/10280321>
- [54] P. Verma, “A Brief Study of Middleware Technologies: Programming Applications and Management Systems”, *Novel Research Aspects in Mathematical and Computer Science Vol. 1*, pp. 174–181, abr. 2022, doi: 10.9734/BPI/NRAMCS/V1/16138D.

- [55] Z. Ereiz y D. Music, "Scrum Without a Scrum Master", *2019 IEEE International Conference on Computer Science and Educational Informatization, CSEI 2019*, pp. 325–328, ago. 2019, doi: 10.1109/CSEI47661.2019.8938877.
- [56] K. Beck *et al.*, "Manifiesto por el Desarrollo Ágil de Software", 2001, Consultado: el 9 de julio de 2024. [En línea]. Disponible en: <https://agilemanifesto.org/iso/es/manifiesto.html>
- [57] W. I. Rochina Rea, "Desarrollo de una guía metodológica para la gestión de proyectos de software con la herramienta GitHub y el marco de trabajo SCRUM.", jul. 2023, Consultado: el 13 de junio de 2024. [En línea]. Disponible en: <https://repositorio.utn.edu.ec/handle/123456789/14598>
- [58] 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", mar. 2024, Consultado: el 13 de junio de 2024. [En línea]. Disponible en: <https://repositorio.utn.edu.ec/handle/123456789/15812>
- [59] K. Schwaber y J. Sutherland, "Scrum Guide | Scrum Guides", 2020, Consultado: el 13 de junio de 2024. [En línea]. Disponible en: <https://scrumguides.org/scrum-guide.html#scrum-artifacts>
- [60] E. A. Vásconez Endara, "Elaboración de una guía metodológica para la gestión de proyectos de software utilizando la herramienta Gitlab y la metodología Scrum para fortalecer los proyectos de desarrollo de software de los estudiantes de la carrera de Ingeniería de Software.", may 2022, Consultado: el 13 de junio de 2024. [En línea]. Disponible en: <https://repositorio.utn.edu.ec/handle/123456789/12443>
- [61] Tips Empresariales, "Emprendimiento | Agile y Scrum - Tips Empresariales", 2020, Consultado: el 13 de junio de 2024. [En línea]. Disponible en: <https://www.tipsempresariales.com/tips/emprendimiento-agile-y-scrum>
- [62] C. Gao, W. Luo, J. Wang, y Y. Zhang, "Software Quality Evaluation Model based on Multiple Linear Regression and Fuzzy Comprehensive Evaluation Method", *Proceedings - 2022 9th*

- International Conference on Dependable Systems and Their Applications, DSA 2022*, pp. 383–389, 2022, doi: 10.1109/DSA56465.2022.00058.
- [63] ISO 25000, “NORMAS ISO 25000”, 2022, Consultado: el 5 de julio de 2024. [En línea]. Disponible en: <https://iso25000.com/index.php/normas-iso-25000>
- [64] S. Karnouskos, R. Sinha, P. Leitao, L. Ribeiro, y T. I. Strasser, “Assessing the Integration of Software Agents and Industrial Automation Systems with ISO/IEC 25010”, *Proceedings - IEEE 16th International Conference on Industrial Informatics, INDIN 2018*, pp. 61–66, sep. 2018, doi: 10.1109/INDIN.2018.8471951.
- [65] M. Kurosu, “Usability, quality in use and the model of quality characteristics”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9169, pp. 227–237, 2015, doi: 10.1007/978-3-319-20901-2_21/FIGURES/3.
- [66] ISO/IEC, “ISO/IEC 25000:2014(en), Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE”, 2014, Consultado: el 14 de junio de 2024. [En línea]. Disponible en: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:25000:ed-2:v1:en>
- [67] ISO/IEC, “ISO/IEC 25022:2016 - Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Measurement of quality in use”, 2016, Consultado: el 14 de junio de 2024. [En línea]. Disponible en: <https://www.iso.org/standard/35746.html>
- [68] W. R. Pradanita y S. Rochimah, “Quality in use of digital wallet based on iso/iec 25022”, *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, vol. 2020-October, pp. 282–286, oct. 2020, doi: 10.23919/EECSI50503.2020.9251300.

- [69] “ISO/IEC 14598-1:1999 - Information technology — Software product evaluation — Part 1: General overview”. Consultado: el 31 de enero de 2025. [En línea]. Disponible en: <https://www.iso.org/standard/24902.html>
- [70] R. Malhotra, D. Kumar, y D. P. Gupta, “An Android Application for Campus Information System”, *Procedia Comput Sci*, vol. 172, pp. 863–868, ene. 2020, doi: 10.1016/J.PROCS.2020.05.124.
- [71] H. Ünlü, D. E. Kennouche, G. K. Soyly, y O. Demirörs, “Microservice-based projects in agile world: A structured interview”, *Inf Softw Technol*, vol. 165, p. 107334, ene. 2024, doi: 10.1016/J.INFSOF.2023.107334.
- [72] M. Cohn, “Agile Estimating and Planning - Mike Cohn - Google Libros”. Consultado: el 17 de octubre de 2024. [En línea]. Disponible en: https://books.google.com.ec/books/about/Agile_Estimating_and_Planning.html?id=BuFWHffRJssC&redir_esc=y
- [73] “ISO/IEC 25022:2016 - Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Measurement of quality in use”. Consultado: el 13 de diciembre de 2024. [En línea]. Disponible en: <https://www.iso.org/standard/35746.html>
- [74] A. Joshi, S. Kale, S. Chandel, y D. K. Pal, “Likert Scale: Explored and Explained”, vol. BJASt, núm. 4, p. 157, 2015, doi: 10.9734/BJAST/2015/14975.
- [75] J. Brooke, “SUS - A quick and dirty usability scale”, 1986, Consultado: el 2 de enero de 2025. [En línea]. Disponible en: <https://rickvanderzwet.nl/trac/personal/export/104/liacs/hci/docs/SUS-questionnaire.pdf>

ANEXOS

Anexo A. Fotografías de aplicación del taller práctico y encuesta SUS



Anexo B. Taller práctico para evaluar la aplicación móvil

INDICACIONES:

1	Complete la tabla con la hora de inicio y la hora de fin de cada tarea
3	Indique si la aplicación mostró algún error durante la tarea (SI/NO)
4	Indique si completó la tarea con éxito (SI/NO)
Nota:	Inicio cada objetivo desde la pantalla principal (Calendario de actividades)

Nivel de carrera:

Objetivo	Nro.	Descripción de la tarea	Nota: presione (Ctrl + h) para insertar la hora			Observación	
			Inicio	Fin	Total		
Enviar una actividad escaneando un documento	1	Buscar y seleccionar la actividad en el Calendario de Actividades			0:00:00		
	2	Llenar el campo descripción			0:00:00		
	3	Escanear un documento como adjunto en la opción Escanear documento y entregar tarea			0:00:00		
Ver la nota de trabajos de la primera parcial de una materia	4	Seleccionar la materia en el Registro Académico			0:00:00		
	5	Ingresar al apartado de Notas			0:00:00		
	6	Seleccionar la primera parcial y expandir el detalle de Trabajos			0:00:00		
Ver las horas asistidas a clases en un día específico de una materia	7	Seleccionar la materia en el Registro Académico			0:00:00		
	8	Ingresar al apartado Más y luego Asistencia			0:00:00		
	9	Seleccionar un día del mes de diciembre			0:00:00		
Solicitar certificado de matrícula	10	Ingresar al menú Más y después Documentos			0:00:00		
	11	Seleccionar y completar la solicitud del certificado deseado			0:00:00		
	12	Revisar el estado de la solicitud en el apartado Historial de Certificados			0:00:00		
Descargar un recurso de una materia	13	Seleccionar la materia en el Registro Académico			0:00:00		
	14	Ingresar al apartado de Recursos			0:00:00		
	15	Desplegar la primera carpeta y descargar un recurso			0:00:00		
Ver los certificados de evento generados en el año 2024	16	Seleccionar el menú Más y luego Documentos			0:00:00		
	17	Seleccionar la opción Eventos			0:00:00		
	18	Definir el rango de búsqueda de Enero 2024 a Diciembre 2024			0:00:00		
Modificar entrega de tarea (Segundo intento)	19	Buscar y seleccionar la actividad en el Calendario de Actividades			0:00:00		
	20	Cambiar el campo descripción			0:00:00		
	21	Eliminar el archivo adjunto existente			0:00:00		
	22	Tomar una fotografía usando la opción Tomar foto y entregar tarea			0:00:00		
Final						Total	0:00:00

Una vez finalizado el taller guárdelo y complete la encuesta.
<https://forms.office.com/r/Qd455MNYC>

Anexo C. Tabulación del taller práctico

id	Facultad	Carrera	Nivel	A1.1	A1.2	A1.3	A2.1	A2.2	A2.3	A3.1	A3.2	A3.3	A4.1	A4.2	A4.3	A5.1	A5.2	A5.3	A6.1	A6.2	A6.3	A7.1	A7.2	A7.3	A7.4	Errores	Tareas Logradas	Objetivos Logrados	Tiempo	Reclamos
1	FICA	Software	2	0:06:22	0:00:18	0:01:14	0:00:02	0:00:06	0:00:05	0:00:04	0:00:05	0:00:06	0:00:03	0:00:10	0:00:08	0:00:21	0:00:06	0:00:09	0:00:04	0:00:04	0:00:17	0:00:04	0:00:21	0:00:13	0:00:24	0	22	7	0:10:46	SI
2	FICA	Software	2	0:06:07	0:00:08	0:01:58	0:00:18	0:00:05	0:00:12	0:00:57	0:00:06	0:00:06	0:00:05	0:00:48	0:00:13	0:00:21	0:00:13	0:00:15	0:00:37	0:00:14	0:00:31	0:00:30	0:00:03	0:00:06	0:00:20	0	22	7	0:14:13	NO
3	FICA	Software	2	0:01:39	0:00:17	0:00:31	0:00:16	0:00:12	0:00:17	0:00:23	0:00:34	0:00:18	0:00:48	0:00:31	0:00:33	0:00:12	0:00:05	0:00:09	0:00:15	0:00:05	0:00:13	0:00:15	0:00:14	0:00:08	0:00:47	0	22	7	0:08:42	SI
4	FICA	Software	2	0:00:39	0:00:11	0:01:26	0:01:00	0:00:20	0:00:06	0:00:13	0:00:22	0:00:06	0:00:47	0:00:44	0:00:07	0:00:11	0:00:09	0:00:12	0:00:12	0:00:21	0:00:58	0:00:12	0:00:33	0:00:07	0:00:15	0	22	7	0:09:11	NO
5	FICA	Software	2	0:00:08	0:00:18	0:04:57	0:00:02	0:00:15	0:00:16	0:00:08	0:00:19	0:00:23	0:00:02	0:00:24	0:00:16	0:00:02	0:00:12	0:00:57	0:00:09	0:00:01	0:00:23	0:00:04	0:00:02	0:00:06	0:00:27	0	22	7	0:09:51	NO
6	FICA	Software	2	0:00:15	0:00:04	0:00:40	0:00:07	0:00:06	0:00:21	0:00:09			0:00:08	0:00:10	0:00:09	0:00:11	0:00:06	0:00:19	0:00:09		0:00:23	0:00:05	0:00:14	0:00:17	0:00:31	0	18	5	0:04:01	SI
7	FICA	Software	2	0:00:18	0:00:03	0:00:09	0:00:17	0:00:12	0:00:16	0:00:15	0:00:05	0:00:07	0:01:02	0:00:29	0:00:38	0:00:13	0:00:07	0:00:08	0:00:23	0:00:10	0:00:25	0:00:29	0:00:21	0:00:07	0:00:41	0	22	7	0:06:55	NO
8	FICA	Software	2	0:00:02	0:00:52	0:00:20	0:00:25	0:00:05	0:00:14	0:00:34	0:00:03	0:00:07	0:00:08	0:00:18	0:00:59	0:00:14	0:00:02	0:00:14	0:00:06	0:00:18	0:00:03	0:00:03	0:00:03	0:00:52	0	22	7	0:06:03	SI	

18	FICA	Software	2	0:00:15	0:00:08	0:00:02	0:00:07	0:00:06	0:00:21	0	18	5	0:04:01	SI
17	FICA	Software	2	0:00:08	0:00:02	0:00:02	0:00:41	0:01:56	0:00:04	0	22	7	0:08:01	NO
16	FICA	Software	2	0:02:46	0:01:37	0:01:20	0:01:36	0:00:13	0:00:37	0	22	7	0:14:36	NO
15	FICA	Software	2	0:00:02	0:00:02	0:03:15	0:00:03	0:00:01	0:00:01	0	22	7	0:04:18	NO
14	FICA	Software	2	0:00:05	0:00:17	0:01:05	0:00:22	0:00:10	0:00:17	0	22	7	0:07:13	NO
13	FICA	Software	2	0:00:13	0:00:11	0:00:18	0:00:06	0:00:09	0:00:10	0	22	7	0:03:21	NO
12	FICA	Software	2	0:06:35	0:00:29	0:00:01	0:00:07	0:00:05	0:00:06	0	22	7	0:10:28	SI
11	FICA	Software	2	0:00:06	0:00:04	0:00:06	0:00:16	0:00:10	0:00:15	0	22	7	0:06:13	NO
10	FICA	Software	2	0:02:03	0:00:04	0:00:03	0:00:03	0:00:02	0:00:01	0	22	7	0:04:24	SI
9	FICA	Software	2	0:00:34	0:00:11	0:00:44	0:00:16	0:00:05	0:00:16	0	22	7	0:05:28	NO

19	FICA	FICA	Software	2	0:00:17	0:00:06	0:00:16	0:00:08	0:00:05	0:00:18	0:00:10	0:00:19	0:00:13	0:00:14	0:00:04	0:00:14	0:01:20	0:00:27	0:00:06	0:00:05	0:00:08	0:00:07	0:00:15	0:00:12	0:00:04	0:00:04	0:01:59	0	22	7	0:07:06	NO	
20	FICA	FICA	Software	2	0:00:12	0:00:13	0:00:29	0:00:12	0:00:07	0:00:09	0:00:05	0:00:05	0:00:04	0:00:07	0:00:04	0:00:17	0:00:07	0:00:04	0:00:04	0:00:05	0:00:08	0:00:07	0:00:17	0:00:12	0:00:09	0:00:04	0:00:04	0:00:25	0	22	7	0:03:51	NO
21	FICA	FICA	Software	2	0:00:06	0:00:05	0:00:29	0:00:08	0:00:08	0:00:18	0:00:07	0:00:08	0:00:09	0:00:09	0:00:21	0:00:28	0:00:05	0:00:09	0:00:04	0:00:05	0:00:08	0:00:07	0:00:20	0:00:07	0:00:13	0:00:06	0:00:06	0:00:30	0	22	7	0:04:31	NO
22	FICA	FICA	Software	2	0:00:27	0:00:18	0:00:55	0:00:22	0:00:34	0:00:21	0:00:19	0:00:25	0:00:18	0:00:23	0:00:18	0:00:50	0:00:21	0:00:14	0:00:14	0:00:25	0:00:18	0:00:34	0:00:22	0:00:10	0:00:10	0:00:35	0:00:03	0	22	7	0:08:34	NO	
23	FICA	FICA	Software	2	0:00:08	0:00:23	0:01:48	0:00:21	0:00:04	0:00:12	0:00:07	0:00:17	0:00:10	0:00:13	0:00:26	0:01:19	0:00:14	0:00:22	0:00:22	0:00:10	0:00:17	0:00:08	0:00:23	0:00:08	0:00:15	0:00:06	0:00:41	0	22	7	0:08:22	NO	
24	FICA	FICA	Software	6	0:00:11	0:00:13	0:01:10	0:00:18	0:00:42	0:00:16	0:00:11	0:00:08	0:00:13	0:00:10	0:00:28	0:04:07	0:00:23	0:00:11	0:00:11	0:00:08	0:00:23	0:00:09	0:00:47	0:00:09	0:00:11	0:00:07	0:00:24	0	22	7	0:10:43	SI	
25	FICA	FICA	Software	6	0:00:48	0:00:30	0:00:35	0:00:15	0:00:29	0:00:51	0:00:28	0:00:11	0:00:26	0:00:14	0:00:21	0:00:12	0:00:09	0:00:13	0:00:16	0:00:21	0:00:11	0:00:23	0:00:22	0:01:03	0:00:23	0:00:56	0:00:30	0	22	7	0:09:56	NO	
26	FICA	FICA	Software	6	0:00:04	0:00:19	0:00:37	0:00:28	0:00:24	0:00:35	0:00:52	0:00:03	0:00:08	0:00:22	0:00:11	0:00:06	0:00:22	0:00:13	0:00:12	0:00:03	0:00:15	0:00:02	0:00:15	0:00:02	0:00:02	0:00:07	0:00:30	0	22	7	0:05:57	NO	
27	FICA	FICA	Software	6	0:00:08	0:00:53	0:00:16	0:00:22	0:00:36	0:00:13	0:00:08	0:00:12	0:00:24	0:00:06	0:00:20	0:00:16	0:00:05	0:00:34	0:00:20	0:00:23	0:00:29	0:01:12	0:00:06	0:00:06	0:00:26	0:00:37	0	22	7	0:08:22	SI		
28	FICA	Software	6	0:00:25	0:00:19	0:00:44	0:00:13	0:00:22	0:00:22	0:00:16	0:00:16	0:00:12	0:00:14	0:00:38	0:00:28	0:00:26	0:00:10	0:00:07	0:00:05	0:00:24	0:00:16	0:00:23	0:00:29	0:00:15	0:00:11	0:00:11	0:00:43	0	22	7	0:06:56	NO	

29	FICA	FICA	Software	6	0:00:13	0:00:03	0:00:19	0:00:43	0:00:19	0:01:02	0:00:51	0:00:20	0:00:10	0:00:10	0:00:19	0:00:46	0:00:15	0:00:21	0:00:05	0:01:01	0:00:07	0:00:01	0:00:30	0:00:31	0:00:10	0:00:08	0:01:30	0	22	7	0:09:51	SI	
30	FICA	FICA	Software	6	0:00:03	0:00:27	0:00:50	0:00:49	0:00:14	0:00:19	0:00:17	0:00:26	0:00:08	0:00:23	0:00:17	0:00:24	0:00:07	0:00:17	0:00:05	0:00:21	0:00:11	0:00:08	0:00:16	0:00:05	0:00:24	0:00:10	0:00:34	0:00:57	0	22	7	0:07:57	NO
31	FICA	FICA	Software	6	0:01:14	0:00:26	0:03:19	0:00:13	0:00:06	0:00:22	0:00:23	0:00:03	0:00:08	0:00:16	0:00:21	0:00:40	0:00:22	0:00:18	0:00:10	0:00:08	0:00:06	0:00:43	0:00:21	0:00:03	0:00:03	0:00:03	0:01:18	0	22	7	0:11:03	NO	
32	FICA	FICA	Software	6	0:05:13	0:00:01	0:00:17	0:00:02	0:00:05	0:00:05	0:00:02	0:00:02	0:00:04	0:00:06	0:00:02	0:00:03	0:00:05	0:00:03	0:00:03	0:00:03	0:00:02	0:00:03	0:00:04	0:00:04	0:00:10	0:00:03	0:00:03	0:00:54	0	22	7	0:07:33	SI
33	FICA	FICA	Software	6	0:01:27	0:00:02	0:00:05	0:00:09	0:00:12	0:00:43	0:00:19	0:00:09	0:00:10	0:00:11	0:00:33	0:01:11	0:00:12	0:00:09	0:01:22	0:00:08	0:00:12	0:00:36	0:00:13	0:00:16	0:00:16	0:00:09	0:00:33	0	22	7	0:09:01	SI	
34	FICA	FICA	Software	6	0:03:19	0:00:03	0:00:03	0:00:32	0:00:05	0:00:34	0:00:12	0:00:08	0:01:04	0:00:09	0:00:17	0:00:03	0:00:16	0:00:04	0:00:34	0:00:01	0:00:04	0:00:02	0:00:07	0:00:06	0:00:06	0:00:02	0:00:12	0	22	7	0:07:57	SI	
35	FICA	FICA	Software	6	0:05:00	0:01:00	0:02:00	0:01:00	0:01:00	0:01:00	0:01:00	0:01:00	0:01:00	0:01:00	0:01:00	0:02:00	0:01:00	0:00:30	0:01:00	0:00:30	0:01:29	0:02:00	0:00:50	0:01:00	0:01:00	0:01:00	0:01:00	0	22	7	0:28:19	NO	
36	FICA	FICA	Software	6	0:01:36	0:01:36	0:02:39	0:01:24	0:00:20	0:00:36	0:00:37	0:00:27	0:01:02	0:00:22	0:00:27	0:00:36	0:00:06	0:00:07	0:02:40	0:02:06	0:04:06	0:00:31	0:00:24	0:00:12	0:00:21	0:00:09	0:00:09	1	22	7	0:22:24	SI	
37	FICA	FICA	Software	6	0:00:22	0:00:23	0:01:31	0:00:16	0:00:28	0:01:40	0:00:24	0:00:39	0:00:35	0:00:08	0:00:14	0:00:11	0:00:40	0:00:11	0:00:19	0:00:06	0:00:04	0:00:27	0:00:13	0:00:06	0:00:24	0:00:27	0	22	7	0:09:48	SI		
38	FICA	Software	6	0:00:11	0:00:13	0:01:10	0:00:18	0:00:42	0:00:16	0:00:11	0:00:08	0:00:08	0:00:10	0:00:28	0:00:09	0:04:07	0:00:23	0:00:11	0:00:11	0:00:11	0:00:04	0:00:47	0:00:09	0:00:11	0:00:07	0:00:24	0	22	7	0:10:43	NO		

39	FICA	FICA	Software	6	0:01:55	0:00:24	0:00:03	0:00:13	0:02:58	0:00:16	0:00:27	0:00:07	0:00:16	0:00:05	0:00:03	0:01:11	0:00:13	0:00:10	0:00:01	0:00:01	0:00:16	0:00:02	0:00:01	0:00:18	0:00:08	0:00:01	0:00:31	1	22	7	0:09:16	SI		
40	FICA	FICA	Software	6	0:00:22	0:00:23	0:00:35	0:00:21	0:00:23	0:00:16	0:00:14	0:00:13	0:00:18	0:00:17	0:00:07	0:01:02	0:00:41	0:00:19	0:00:06	0:00:06	0:00:23	0:00:11	0:00:09	0:01:21	0:00:36	0:00:12	0:00:08	0:00:15	0:00:27	1	22	7	0:09:10	SI
41	FICA	FICA	Software	6	0:08:45	0:01:15	0:00:50	0:00:51	0:00:38	0:01:06	0:00:41	0:00:35	0:00:21	0:00:37	0:00:01	0:02:02	0:02:26	0:02:24	0:01:03	0:02:31	0:01:01	0:02:59	0:00:27	0:00:13	0:00:06	0:00:24	0:00:27	0	22	7	0:09:48	SI		
42	FICA	FICA	Software	6	0:00:15	0:00:18	0:00:37	0:00:21	0:00:04	0:00:12	0:00:08	0:00:05	0:00:10	0:00:08	0:00:05	0:02:02	0:02:26	0:02:24	0:01:03	0:02:31	0:01:01	0:02:59	0:00:27	0:00:13	0:00:06	0:00:32	0:00:09	1	22	7	0:33:00	SI		
43	FICA	FICA	Software	8	0:00:36	0:00:07	0:00:52	0:00:09	0:00:07	0:00:12	0:00:08	0:00:07	0:00:11	0:00:15	0:00:13	0:00:09	0:00:06	0:00:10	0:00:17	0:00:10	0:00:04	0:00:27	0:00:03	0:00:16	0:00:16	0:00:17	0:00:05	0:00:40	0	22	7	0:04:44	NO	
44	FICA	FICA	Software	8	0:00:49	0:00:18	0:00:44	0:00:38	0:00:29	0:00:09	0:00:12	0:00:22	0:00:19	0:00:21	0:02:00	0:00:13	0:00:12	0:00:03	0:00:08	0:01:02	0:00:29	0:00:12	0:00:13	0:00:37	0:00:09	0:00:21	0:00:20	0	22	7	0:10:11	NO		
45	FICA	FICA	Software	8	0:00:15	0:00:29	0:00:20	0:00:11	0:00:10	0:00:16	0:00:07	0:00:17	0:00:13	0:00:05	0:01:03	0:00:17	0:00:09	0:00:05	0:00:22	0:00:04	0:00:18	0:00:17	0:00:11	0:00:11	0:00:25	0:01:08	0	22	7	0:06:45	NO			
46	FICA	FICA	Software	8	0:00:37	0:02:30	0:00:13	0:00:27	0:00:23	0:00:26	0:00:53	0:00:23	0:00:22	0:00:19	0:00:21	0:00:15	0:00:17	0:00:09	0:00:22	0:00:03	0:00:18	0:00:13	0:00:39	0:00:07	0:00:26	0:00:12	0	22	7	0:09:57	NO			
47	FICA	FICA	Software	7	0:00:12	0:00:04	0:00:39	0:00:08	0:00:14	0:00:14	0:00:08	0:00:08	0:00:06	0:00:06	0:00:08	0:00:08	0:00:09	0:00:05	0:00:22	0:00:03	0:00:04	0:00:18	0:00:17	0:00:11	0:00:25	0:01:08	0	22	7	0:04:18	NO			
48	FICA	FICA	Software	7	0:00:12	0:00:04	0:00:39	0:00:08	0:00:14	0:00:14	0:00:08	0:00:08	0:00:06	0:00:06	0:00:08	0:00:08	0:00:09	0:00:05	0:00:22	0:00:03	0:00:04	0:00:18	0:00:17	0:00:11	0:00:25	0:01:08	0	22	7	0:04:18	NO			

49	FICA	FICA	Software	∞	∞	0:00:07	0:00:07	0:00:19	0:00:36	0:00:19	0:00:02	0:00:02	0:00:19	0:00:20	0:00:05	0:00:06	0:00:04	0:00:13	0:00:11	0:00:06	0:00:03	0:00:06	0:00:07	0:00:05	0:00:21	0:00:08	0:00:14	0:00:06	0:00:34	0	22	7	0:04:31	NO	
50	FICA	FICA	Software	∞	∞	0:00:22	0:00:22	0:00:07	0:03:33	0:00:02	0:00:04	0:00:04	0:00:28	0:00:05	0:00:02	0:00:36	0:00:07	0:01:00	0:00:17	0:00:07	0:00:05	0:00:09	0:00:09	0:00:12	0:00:02	0:00:11	0:00:23	0:00:04	0:00:08	0:00:11	0	22	7	0:08:15	NO
51	FICA	FICA	Software	∞	∞	0:00:37	0:00:37	0:02:30	0:00:13	0:00:27	0:00:23	0:00:26	0:00:53	0:00:23	0:00:22	0:00:19	0:00:21	0:00:15	0:00:16	0:00:09	0:00:09	0:00:11	0:00:17	0:00:18	0:00:13	0:00:39	0:00:07	0:00:26	0:00:12	0	22	7	0:09:57	NO	
52	FICA	FICA	Software	∞	∞	0:00:47	0:00:47	0:00:08	0:00:15	0:00:49	0:00:26	0:00:23	0:00:19	0:00:18	0:00:04	0:00:12	0:00:38	0:00:09	0:00:16	0:00:07	0:00:07	0:00:04	0:00:16	0:00:06	0:00:10	0:00:15	0:00:08	0:00:03	0:00:24	0	22	7	0:06:17	NO	
53	FICA	FICA	Software	∞	∞	0:01:17	0:01:17	0:00:08	0:00:13	0:00:10	0:00:08	0:00:21	0:00:06	0:00:15	0:00:11	0:00:16	0:00:32	0:00:09	0:00:07	0:00:05	0:00:12	0:00:18	0:00:12	0:00:29	0:00:14	0:00:14	0:00:14	0:00:08	0:00:48	0	22	7	0:06:33	NO	
54	FICA	FICA	Software	∞	∞	0:00:06	0:00:06	0:00:09	0:01:00	0:00:17	0:00:21	0:00:17	0:00:11	0:00:12	0:00:15	0:00:24	0:00:32	0:00:19	0:00:07	0:00:07	0:00:09	0:00:09	0:00:14	0:00:08	0:00:34	0:00:18	0:00:12	0:00:37	0	22	7	0:06:44	NO		
55	FICA	FICA	Software	∞	∞	0:00:08	0:00:08	0:00:13	0:00:21	0:00:16	0:00:10	0:00:11	0:00:13	0:00:07	0:00:26	0:00:13	0:00:20	0:00:17	0:00:03	0:00:05	0:00:10	0:00:01	0:00:03	0:00:27	0:00:02	0:00:09	0:00:06	0:00:16	0	22	7	0:04:17	NO		
56	FICA	FICA	Software	7	7	0:00:34	0:00:34	0:00:20	0:00:42	0:00:16	0:00:55	0:00:08	0:00:02	0:00:07	0:00:01	0:01:15	0:00:05	0:00:06	0:00:03	0:00:03	0:00:08	0:00:08	0:00:06	0:00:18	0:00:06	0:00:07	0:00:01	0:00:02	0	22	7	0:06:03	NO		
57	FICA	FICA	Software	∞	∞	0:00:35	0:00:35	0:00:08	0:00:09	0:00:25	0:00:39	0:00:49	0:00:24	0:00:23	0:00:02	0:00:20	0:00:39	0:00:21	0:00:25	0:00:25	0:00:25	0:00:48	0:00:49	0:00:48	0:00:37	0:00:37	0:00:37	0:00:37	0	22	7	0:11:34	NO		
58	FICA	FICA	Software	10	10	0:00:46	0:00:46	0:00:32	0:00:55	0:00:08	0:00:06	0:00:26	0:00:03	0:00:18	0:00:34	0:00:02	0:00:03	0:00:15	0:00:03	0:00:06	0:00:06	0:00:06	0:00:18	0:00:06	0:00:18	0:00:37	0:00:37	0:00:37	0	22	7	0:06:02	NO		

66	FICA	Software	∞	0:00:07	0:00:09	0:00:14	0:00:25	0:00:40	0:00:09	0:00:07	0:00:44	0:00:04	0:00:03	0:00:17	0:00:06	0:00:13	0:00:09	0:00:11	0:00:30	0:00:06	0:00:09	0:00:16	0:00:08	0:00:26	0:01:59	0:00:23	0:00:12	0:00:14	0:00:06	0:00:08	0:00:24	0:00:19	0:00:09	0:00:12	0:00:06	0:00:02	0:00:22	0:01:42	0	22	7	0:05:04	NO
65	FICA	Software	∞	0:00:09	0:00:07	0:00:14	0:00:04	0:00:44	0:00:09	0:00:07	0:00:44	0:00:04	0:00:03	0:00:17	0:00:06	0:00:13	0:00:09	0:00:16	0:00:32	0:00:08	0:00:07	0:00:16	0:00:08	0:00:26	0:01:59	0:00:23	0:00:12	0:00:14	0:00:06	0:00:08	0:00:24	0:00:19	0:00:09	0:00:12	0:00:06	0:00:02	0:00:22	0:01:42	0	22	7	0:03:42	NO
64	FICA	Software	∞	0:00:22	0:00:23	0:00:14	0:00:33	0:00:14	0:00:09	0:00:23	0:00:14	0:00:33	0:00:13	0:00:16	0:00:11	0:00:09	0:00:06	0:00:32	0:00:21	0:00:08	0:00:07	0:00:16	0:00:32	0:00:26	0:01:59	0:00:23	0:00:12	0:00:14	0:00:06	0:00:08	0:00:24	0:00:19	0:00:09	0:00:12	0:00:07	0:00:56	0	22	7	0:08:13	NO		
63	FICA	Software	∞	0:01:54	0:00:06	0:02:59	0:00:29	0:02:59	0:00:05	0:00:02	0:00:11	0:00:31	0:00:03	0:00:11	0:00:31	0:00:03	0:00:27	0:00:10	0:00:31	0:00:03	0:00:03	0:00:04	0:00:04	0:00:03	0:00:11	0:00:04	0:00:11	0:00:03	0:00:03	0:00:10	0:00:19	0:00:09	0:00:11	0:00:02	0:01:42	1	22	7	0:12:44	SI			
62	FICA	Software	∞	0:00:08	0:00:07	0:00:35	0:00:19	0:00:35	0:00:09	0:00:17	0:00:23	0:00:11	0:00:23	0:00:11	0:00:41	0:00:43	0:00:11	0:00:10	0:00:32	0:00:17	0:00:07	0:00:17	0:00:27	0:00:26	0:02:18	0:00:41	0:00:43	0:00:06	0:00:22	0:00:11	0:00:10	0:00:32	0:00:17	0:00:48	0	22	7	0:09:13	NO				
61	FICA	Software	∞	0:00:20	0:00:17	0:01:38	0:00:43	0:01:38	0:00:17	0:00:53	0:00:30	0:00:09	0:00:30	0:00:43	0:00:54	0:00:49	0:00:14	0:00:26	0:00:43	0:00:30	0:00:05	0:00:30	0:00:43	0:01:47	0:01:03	0:00:54	0:00:49	0:00:13	0:00:23	0:00:14	0:00:26	0:00:43	0:00:23	0:01:10	0	22	7	0:12:55	NO				
60	FICA	Software	∞	0:00:03	0:00:17	0:01:16	0:00:10	0:01:16	0:00:18	0:00:10	0:00:04	0:00:27	0:00:10	0:00:18	0:00:11	0:00:10	0:00:11	0:00:05	0:00:38	0:00:05	0:00:05	0:00:10	0:00:27	0:01:16	0:01:03	0:00:11	0:00:10	0:00:03	0:00:02	0:00:11	0:00:05	0:00:18	0:00:38	0:00:04	0:01:28	0	22	7	0:06:34	SI			
59	FICA	Software	∞	0:00:34	0:00:20	0:00:42	0:00:16	0:00:42	0:00:55	0:00:08	0:00:02	0:00:07	0:00:01	0:00:15	0:00:05	0:00:33	0:00:06	0:00:03	0:00:08	0:00:01	0:00:08	0:00:07	0:01:15	0:00:05	0:00:33	0:00:06	0:00:03	0:00:08	0:00:08	0:00:06	0:00:18	0:00:06	0:00:01	0:00:02	0	22	7	0:06:03	NO				

Anexo D. Tabulación encuesta SUS

Id	Facultad	Carrera	Nivel	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	FICA	Software	2	5	1	5	2	5	2	5	1	5	1
2	FICA	Software	2	5	4	5	1	4	1	5	1	5	1
3	FICA	Software	2	5	2	4	1	4	3	5	1	5	1
4	FICA	Software	2	5	4	5	4	5	2	4	2	4	2
5	FICA	Software	2	5	1	4	4	5	2	5	4	5	1
6	FICA	Software	2	4	2	4	1	4	2	4	1	4	1
7	FICA	Software	2	5	2	3	1	4	2	5	1	5	1
8	FICA	Software	2	5	2	5	3	4	2	5	2	3	1
9	FICA	Software	2	5	1	4	1	5	1	5	1	3	1
10	FICA	Software	2	4	3	5	1	5	2	5	2	4	3
11	FICA	Software	2	5	1	5	1	5	1	5	1	5	1
12	FICA	Software	2	5	2	4	4	5	2	5	1	4	2
13	FICA	Software	2	5	3	4	3	4	3	4	3	4	1
14	FICA	Software	2	5	1	5	1	5	1	5	1	5	1
15	FICA	Software	2	4	2	4	3	5	3	5	2	3	2
16	FICA	Software	2	5	3	5	5	5	1	5	1	5	2
17	FICA	Software	2	5	3	5	5	5	4	5	5	5	5
18	FICA	Software	2	5	1	5	1	5	1	5	3	5	2
19	FICA	Software	2	5	4	5	1	4	3	5	5	4	4
20	FICA	Software	2	5	4	5	1	5	1	5	1	5	1
21	FICA	Software	2	5	1	5	1	5	1	5	1	5	1
22	FICA	Software	2	1	2	3	4	3	2	3	4	2	3
23	FICA	Software	2	4	4	3	1	5	1	4	4	4	1
24	FICA	Software	6	5	2	4	1	5	1	4	1	5	1
25	FICA	Software	6	4	3	4	1	4	1	4	1	5	1
26	FICA	Software	6	5	2	4	1	5	1	4	1	2	2

27	FICA	Software	6	5	1	4	1	5	1	5	4	5	1
28	FICA	Software	6	5	1	5	1	5	1	4	1	5	2
29	FICA	Software	6	5	2	4	1	4	1	5	1	4	1
30	FICA	Software	6	5	1	5	4	4	1	5	1	5	1
31	FICA	Software	6	4	1	5	2	4	1	4	5	5	1
32	FICA	Software	6	5	3	5	3	5	3	5	3	5	2
33	FICA	Software	6	5	4	5	2	5	2	5	5	4	2
34	FICA	Software	6	5	5	5	2	5	2	5	1	4	4
35	FICA	Software	6	4	2	4	1	4	2	5	2	5	1
36	FICA	Software	6	5	2	5	1	4	1	5	1	5	1
37	FICA	Software	6	4	1	4	1	4	1	4	1	5	2
38	FICA	Software	6	5	1	4	1	5	1	4	1	5	1
39	FICA	Software	6	4	3	3	4	4	3	4	3	3	4
40	FICA	Software	6	5	1	5	1	4	2	5	1	5	1
41	FICA	Software	6	4	2	4	4	4	4	4	3	4	3
42	FICA	Software	6	5	2	4	1	5	2	5	2	5	2
43	FICA	Software	8	4	2	4	2	4	2	4	2	4	3
44	FICA	Software	8	5	5	5	1	5	2	5	1	5	1
45	FICA	Software	8	5	1	5	1	5	1	5	1	5	1
46	FICA	Software	8	5	2	5	2	5	1	4	1	5	1
47	FICA	Software	8	5	5	5	5	5	5	5	5	5	5
48	FICA	Software	7	5	1	5	1	5	1	5	1	5	1
49	FICA	Software	8	5	2	5	1	5	1	4	1	5	1
50	FICA	Software	8	5	3	4	3	5	1	5	2	5	2
51	FICA	Software	8	5	5	4	5	5	4	4	5	4	5
52	FICA	Software	8	5	4	5	4	4	4	4	4	4	4
53	FICA	Software	8	4	3	4	3	4	2	4	4	3	2
54	FICA	Software	8	4	2	4	1	5	1	4	1	5	1
55	FICA	Software	8	5	5	5	2	5	2	5	2	5	1
56	FICA	Software	7	4	2	4	1	4	1	5	1	5	1

57	FICA	Software	8	5	3	4	2	5	2	4	2	4	2
58	FICA	Software	10	5	3	4	2	5	2	5	2	5	3
59	FICA	Software	8	5	2	5	1	5	1	5	1	5	1
60	FICA	Software	8	5	2	5	2	4	2	3	1	4	1
61	FICA	Software	8	5	1	5	1	5	1	5	1	3	5
62	FICA	Software	8	5	5	5	1	5	1	5	1	3	1
63	FICA	Software	8	4	1	5	2	5	1	5	3	5	1
64	FICA	Software	8	5	1	5	1	5	1	5	1	5	1
65	FICA	Software	8	5	2	5	2	4	2	4	2	5	1
66	FICA	Software	8	5	1	4	1	4	1	4	2	4	1