

UNIVERSIDAD TÉCNICA DEL NORTE



**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE SOFTWARE**

**Implementación del servicio "Notificaciones y eventos" en la
plataforma móvil de la Universidad Técnica del Norte
utilizando el marco de trabajo SCRUM y la norma ISO/IEC
25022**

AUTOR:

Brandon Leonardo Flores Alcucer

DIRECTORA:

Ing. Cathy Pamela Guevara Vega, Ph.D.

**Ibarra- Ecuador
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:	1050258431		
APELLIDOS Y NOMBRES:	Flores Alcucer Brandon Leonardo		
DIRECCIÓN:	Quetzal y Tungurahua, Ibarra		
EMAIL:	blforesa@utn.edu.ec / alcucerbrandon@gmail.com		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0984918238

DATOS DE LA OBRA	
TÍTULO:	Implementación del servicio "Notificaciones y eventos" en la plataforma móvil de la Universidad Técnica del Norte utilizando el marco de trabajo SCRUM y la norma ISO/IEC 25022.
AUTOR (ES):	Brandon Leonardo Flores Alcucer
FECHA DE APROBACIÓN: DD/MM/AAAA	5 de septiembre de 2025
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	INGENIERO EN SOFTWARE
ASESOR /DIRECTOR:	Ing. Cathy Pamela Guevara Vega, 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 esta y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 5 días del mes de septiembre de 2025

EL AUTOR:



.....
Nombre: Flores Alcucer Brandon Leonardo

CI: 1050258431

CERTIFICACIÓN DEL DIRECTOR

Certifico que el trabajo de grado “Implementación del servicio Notificaciones y Eventos en la plataforma móvil de la Universidad Técnica del Norte utilizando el marco de trabajo SCRUM y la norma ISO/IEC 25022”, ha desarrollado en su totalidad por el señor: Brandon Leonardo Flores Alcucer, portador de la cédula de identidad número 1050258431.

.....

Ing. Cathy Pamela Guevara Vega, Ph.D.

Directora del Trabajo de Grado

DEDICATORIA

A mis amados padres, Fernando Flores y Marina Alcucer, quienes fueron, son y siempre serán un pilar fundamental en mi vida. Por apoyarme incondicionalmente tanto en lo personal como en mi formación académica y profesional. En cada circunstancia, favorable o desfavorable, supieron sacar adelante no solo a mí, sino también a mis hermanos. Agradezco con orgullo desde el fondo de mi corazón por tener unos padres que desde siempre me han inculcado los valores correctos para seguir adelante: el respeto, la responsabilidad, la honestidad y, por sobre todas las cosas, la humildad.

A mis queridos hermanos, Erick Flores y Fernando Flores, quienes siempre supieron indicarme el camino correcto a través de su ejemplo y por protegerme en todo momento. Desde que nací han sido mis segundos tutores, acompañándome a lo largo de toda mi formación.

A ustedes les dedico este gran logro, por haber sido parte esencial de un nuevo paso hacia adelante en mi vida.

AGRADECIMIENTO

A mis padres, por apoyarme incondicionalmente a lo largo de toda mi vida y por estar siempre presentes en cada uno de mis objetivos. Su sacrificio y amor inquebrantable hacia sus hijos me han dado la fuerza necesaria para alcanzar cada meta propuesta. Les agradezco profundamente por enseñarme que cumplir un sueño implica aprender a caer, pero sobre todo a levantarse, manteniéndome siempre fiel a mis principios, valores y con humildad al alcanzar cualquier logro.

A mis hermanos, por acompañarme y brindarme apoyo en momentos difíciles, por enseñarme el verdadero significado del lazo fraterno y por estar siempre ahí cuando más los necesitaba. Gracias por ser parte fundamental en este nuevo paso de mi vida.

A mis tutores, Cathy Guevara y Antonio Quiña, gracias por confiar en mí y darme la oportunidad de demostrar de lo que soy capaz en este gran proyecto de mi vida. Su guía, dedicación, conocimientos y compromiso han sido clave para alcanzar este sueño.

De igual forma, extiendo mi gratitud a todos los y las docentes de la carrera de Ingeniería en Software, desde el primer nivel hasta el último. Gracias por compartir conmigo parte de su conocimiento, experiencia y vocación, permitiéndome culminar con satisfacción este importante capítulo de mi vida.

Agradezco a la Universidad Técnica del Norte por brindarme una formación integral, no solo en conocimientos técnicos, sino también en valores fundamentales para convertirme en un profesional de calidad.

Tabla de contenido

Planteamiento del Problema	13
Objetivos	14
Objetivo general	14
Objetivos específicos	14
Alcance.....	14
Metodología.....	16
Justificación.....	17
Trabajos Relacionados	17
I. MARCO TEÓRICO	19
1.1 Notificaciones y eventos en la Universidad Técnica del Norte	19
1.1.1 Proceso de notificaciones actual	19
1.1.2 Sistema Informático Integrado Universitario.....	19
1.2 Introducción a las aplicaciones móviles	20
1.2.1 Evolución y tendencias en el desarrollo de aplicaciones móviles.....	20
1.2.2 Importancia de las notificaciones en aplicaciones móviles.....	21
1.3 Arquitectura de microservicios	22
1.3.1 Conceptos y beneficios de la arquitectura de microservicios	22
1.3.2 Implementación de microservicios en aplicaciones móviles.....	22
1.4 Metodología ágil Scrum.....	23
1.4.1 Principios y prácticas de Scrum	23
1.4.2 Roles de SCRUM.....	24
1.4.3 Fases de Scrum.....	24
1.4.4 Personal Process Software (PSP).....	25
1.5 Normas de evaluación de calidad	26
1.5.1 Norma ISO/IEC 25022: Modelos de calidad en uso de sistemas informáticos	27
1.5.2 Evaluación de la calidad en uso de servicios en aplicaciones móviles	27
1.6 Experimento para la selección del lenguaje de programación	28
1.7 Herramientas de desarrollo	29
1.7.1 Node.JS.....	29
1.7.2 Oracle Database	30
1.7.3 Next.js en el desarrollo de aplicaciones web	31
1.7.4 Flutter en el desarrollo de aplicaciones móviles.....	32
1.8 Notificaciones Push	32
1.8.1 Funcionamiento y tipo de notificaciones push	33
1.8.2 Buenas prácticas en el envío de notificaciones.....	33

1.8.3	Firestore Cloud Messaging	33
1.9	Desarrollo de aplicaciones multiplataforma	34
1.9.1	Ventajas y desafíos de aplicaciones multiplataforma	34
1.10	Trabajos Relacionados	34
II.	DESARROLLO	36
2.1	Fase de Pre-juego	36
2.1.1	Equipo Scrum	36
2.1.2	Levantamiento de requisitos	37
2.1.3	Product Backlog	51
2.2	Fase de Juego	52
2.2.1	Planificación Sprint 0	52
2.2.2	Sprint 1	58
2.2.3	Sprint 2	63
2.2.4	Sprint 3	70
2.2.5	Sprint 4	76
III.	VALIDACIÓN	79
3.1	Modelo de calidad en uso	79
3.2	Muestra poblacional	80
3.3	Instrumentos de evaluación	80
3.4	Fiabilidad de los instrumentos	82
3.4.1	Fiabilidad del taller práctico	82
3.4.2	Fiabilidad de la encuesta SUS	83
3.5	Evaluación del modelo de calidad en uso	84
3.5.1	Evaluación de eficacia	84
3.5.2	Evaluación de eficiencia	85
3.5.3	Evaluación de satisfacción	85
3.6	Resultados del modelo de calidad en uso	87
	CONCLUSIONES	89
	RECOMENDACIONES	90
	REFERENCIAS	91
	ANEXOS	97
	Anexo A. Fotografías del taller práctico y encuesta SUS a usuarios	97
	Anexo B. Taller práctico de evaluación	98
	Anexo C. Resultados del taller práctico	99
	Anexo D. Resultados encuesta SUS	100

Índice de Figuras

Figura 1	Árbol de Problemas	13
Figura 2	Arquitectura de la aplicación.....	16
Figura 3	Procesos de Scrum [26].....	24
Figura 4	Organización de la familia de normas ISO/IEC 25000.....	26
Figura 5	Porcentaje de evaluación de lenguajes.....	28
Figura 6.	Arquitectura de FCM [52].....	34
Figura 7	Distribución docentes- Encuesta diagnóstica.....	39
Figura 8	Distribución estudiantes- Encuesta diagnóstica.....	39
Figura 9	Percepción de utilidad docentes.....	41
Figura 10	Percepción de utilidad estudiantes.....	42
Figura 11	Arquitectura general del proyecto.....	53
Figura 12	Arquitectura servicio de notificaciones	54
Figura 13	Diagrama de secuencia	55
Figura 14	Estructura de directorios backend	56
Figura 15	Estructura de directorios aplicación web	56
Figura 16	Estructura de directorios aplicación móvil.....	57
Figura 17	Permisos de notificación en la aplicación.....	60
Figura 18	Servicio para el envío de notificaciones FCM.....	60
Figura 19	Notificación de prueba vía FCM.....	61
Figura 20	Controller suscripción a grupos/topics FCM.....	61
Figura 21	Servicio para la entrega de notificaciones de avance y asistencia	62
Figura 22	Consulta para obtener la materia en trigger	65
Figura 23	Lógica para el envío de notificaciones en cola	65
Figura 24	Variables dentro del trigger de calificaciones	66
Figura 25	Función para notificaciones de horario	66
Figura 26	Buzón de notificaciones (leídas)	67
Figura 27	Buzón de notificaciones (no leídas).....	67
Figura 28	Preferencias de docentes	67
Figura 29	Notificación push en iOS	68
Figura 30	Preferencias de estudiantes.....	68
Figura 31	Formulario web para el envío de notificaciones	72
Figura 32	Vista web de dispositivos asociados a usuarios	72
Figura 33	Vista web de dispositivos de la aplicación UTN-Móvil.....	73
Figura 34	Vista web para gestión de suscripciones.....	73
Figura 35	Vista con la lista de notificaciones.....	74
Figura 36	Modal para la gestión del estado de las notificaciones	74
Figura 37	Vista web para la gestión de grupos/topics	74
Figura 38	Formulario web para programar notificaciones	77
Figura 39	Job programado para la notificación de tareas pendientes.....	78
Figura 40	Función para el envío de notificaciones de tareas pendientes	78
Figura 41	Prueba de normalidad taller práctico	82
Figura 42	Matriz de correlaciones taller práctico	83
Figura 43	Alfa de Cronbach encuesta SUS.....	83
Figura 44	Resultado de calidad en uso [58].....	88

Índice de Tablas

Tabla 1 Trabajos relacionados	17
Tabla 2 Tecnologías seleccionadas para el experimento	28
Tabla 3 Roles Scrum del proyecto	36
Tabla 4 Planificación Sprints	37
Tabla 5 Preguntas docente - Percepción de utilidad en notificaciones	40
Tabla 6 Preguntas estudiante - Percepción de utilidad en notificaciones.....	41
Tabla 7 Historia de Usuario 1	42
Tabla 8 Historia de Usuario 2	43
Tabla 9 Historia de Usuario 3	43
Tabla 10 Historia de Usuario 4	44
Tabla 11 Historia de Usuario 5	44
Tabla 12 Historia de Usuario 6	45
Tabla 13 Historia de Usuario 7	45
Tabla 14 Historio de Usuario 8	46
Tabla 15 Historia de Usuario 9	47
Tabla 16 Historia de Usuario 10	47
Tabla 17 Historia de Usuario 11	48
Tabla 18 Historia de Usuario 12	48
Tabla 19 Historia de Usuario 13	49
Tabla 20 Historia de Usuario 14	49
Tabla 21 Historia de Usuario 15	50
Tabla 22 Historia de Usuario 16	50
Tabla 23 Product Backlog.....	51
Tabla 24 Sprint Backlog 1	59
Tabla 25 Revisión Sprint 1	62
Tabla 26 Sprint Backlog 2	64
Tabla 27 Revisión Sprint 2	69
Tabla 28 Sprint Backlog 3	71
Tabla 29 Revisión Sprint 3	75
Tabla 30 Sprint Backlog 4	77
Tabla 31 Modelo de Calidad en Uso	79
Tabla 32 Preguntas de encuesta SUS.....	80
Tabla 33 Taller práctico basado en ISO/IEC 25022	81
Tabla 34 Peso de respuestas SUS.....	85
Tabla 35 Resultado SUS - Utilidad	86
Tabla 36 Resultados métrica de Comodidad.....	87
Tabla 37 Resultados del Modelo de Calidad en Uso.....	87

RESUMEN

El auge de las plataformas móviles ha transformado la forma en que los usuarios acceden a servicios digitales, exigiendo soluciones optimizadas para dispositivos móviles. En la Universidad Técnica del Norte, el sistema SIIU, disponible principalmente a través de navegadores web, presenta limitaciones para los usuarios que requieren acceso móvil ágil y notificaciones inmediatas sobre eventos académicos. Para resolver esta problemática, en el presente trabajo de tesis se desarrolló e implementó el servicio "Notificaciones y Eventos" en la plataforma UTN-Móvil, utilizando el marco de trabajo Scrum para garantizar un proceso iterativo y centrado en las necesidades del usuario.

El proyecto se enfocó en implementar un sistema de notificaciones push, desarrollando tanto el módulo para la aplicación móvil como un sistema web de administración, con el objetivo de optimizar la comunicación institucional mediante Firebase Cloud Messaging (FCM). La evaluación de calidad se basó en los criterios de la norma ISO/IEC 25022, analizando aspectos como eficacia, eficiencia y satisfacción.

Los resultados demostraron que la solución cumple con los estándares esperados, ofreciendo un rendimiento sólido en la ejecución de tareas y una experiencia de usuario fluida. Este avance facilita la difusión de información académica y refuerza la conexión entre la universidad y su comunidad, asegurando un acceso rápido y eficiente a servicios clave. Además, se identificaron oportunidades para optimizar la confiabilidad de las notificaciones, lo que permitirá futuras mejoras.

Palabras clave: plataformas móviles, notificaciones push, educación superior, Scrum, calidad de software, ISO/IEC 25022, FCM, firebase.

ABSTRACT

The rise of mobile platforms has transformed the way users access digital services, demanding solutions optimized for mobile devices. At the Technical University of the North, the SIIU system, available mainly through web browsers, has limitations for users who require agile mobile access and immediate notifications about academic events. To solve this problem, this project developed and implemented the “Notifications and Events” service on the UTN-Mobile platform, using the Scrum framework to ensure an iterative process focused on user needs.

The project focused on implementing a push notification system, developing both the module for the mobile application and a web administration system, with the aim of optimizing institutional communication through Firebase Cloud Messaging (FCM). The quality assessment was based on the criteria of the ISO/IEC 25022 standard, analyzing aspects such as effectiveness, efficiency, and satisfaction.

The results demonstrated that the solution meets the expected standards, offering solid performance in task execution and a smooth user experience. This advancement facilitates the dissemination of academic information and strengthens the connection between the university and its community, ensuring fast and efficient access to key services. In addition, opportunities to optimize the reliability of notifications were identified, which will enable future improvements.

Keywords: mobile platforms, push notifications, higher education, Scrum, software quality, ISO/IEC 25022, FCM, Firebase.

INTRODUCCIÓN

Planteamiento del Problema

El crecimiento en la adopción tecnológica se evidencia en el aumento constante de usuarios y aplicaciones móviles a nivel global en los últimos años. Por ejemplo, el número de usuarios pasó de 3600 millones en 2016 a 6500 millones en 2022 [1], [2]. Este avance tecnológico también ha impulsado a las instituciones educativas a digitalizar sus servicios y recursos académicos, facilitando así un acceso continuo para estudiantes y docentes [3]. Actualmente, la Universidad Técnica del Norte (UTN) cuenta con un sistema informático integrado llamado "SIIU", ofreciendo diversos servicios para estudiantes y docentes. Sin embargo, el acceso a este sistema está limitado de manera prioritaria a través de un navegador web, lo que dificulta la comodidad de los usuarios que desean acceder desde dispositivos móviles y recibir información instantánea sobre eventos y actividades universitarias. Para abordar esta problemática, se propone en este trabajo la implementación de un servicio de notificaciones y eventos en la plataforma UTN-Móvil, junto con un sistema web dedicado a la gestión de notificaciones de eventos universitarios. Esta iniciativa tiene como objetivo fortalecer la comunicación dentro de la comunidad universitaria y amplificar la difusión de eventos y actividades académicas. La situación actual y los principales problemas se resumen en el siguiente esquema representado en la **Figura 1**.

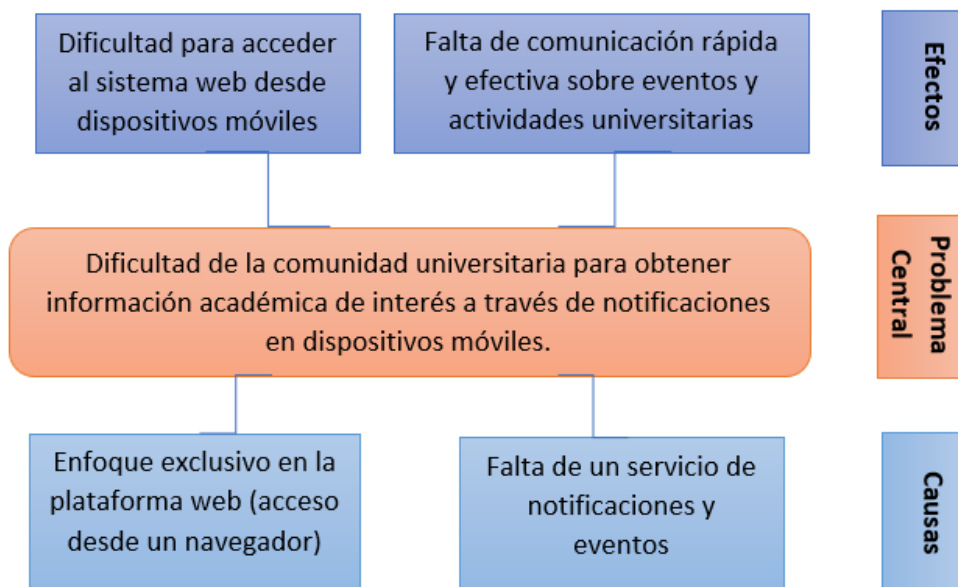


Figura 1 Árbol de Problemas

Fuente: Propia

Objetivos

Objetivo general

Implementar el servicio "Notificaciones y Eventos" en la plataforma móvil de la Universidad Técnica del Norte, utilizando el marco de trabajo SCRUM y cumpliendo con los estándares de calidad definidos por la ISO/IEC 25022.

Objetivos específicos

- Establecer un marco conceptual para el estudio, que proporcione un contexto detallado para la implementación del servicio de "Notificaciones y Eventos".
- Implementar el servicio de "Notificaciones y Eventos" para la plataforma móvil de la UTN, junto con un sistema web para la gestión de notificaciones personalizadas, mediante el uso del marco de trabajo SCRUM.
- Evaluar la calidad en uso del producto de software mediante la aplicación de los criterios establecidos por la norma ISO/IEC 25022.

Alcance

El presente trabajo de titulación tiene como finalidad implementar el servicio de "Notificaciones y Eventos" para la plataforma UTN-Móvil. Este servicio abarca tanto el recibimiento de notificaciones automáticas como notificaciones personalizadas (enviadas desde el sistema web de gestión de notificaciones) a los usuarios de la plataforma móvil.

Notificaciones Automáticas:

Las notificaciones automáticas se activan ante eventos y actividades académicas de interés, determinadas en función de los resultados obtenidos mediante encuestas de diagnóstico y percepción aplicadas a usuarios docentes y estudiantes de la UTN. Se priorizaron aquellas funcionalidades del portafolio académico utilizadas con mayor frecuencia.

Para docentes, los tipos de notificaciones automáticas considerados más útiles incluyen:

- Recordatorio de horario de clases.
- Recordatorio de registro de asistencia pendiente.
- Recordatorio de registro de avance pendiente.

Para estudiantes, se contemplan notificaciones automáticas ante:

- Asignación de nuevas tareas.
- Publicación de calificaciones.
- Recordatorio de tareas próximas a vencerse.

Además, tanto docentes como estudiantes pueden gestionar sus preferencias sobre qué notificaciones desean recibir en sus dispositivos.

Notificaciones Personalizadas:

- Las notificaciones personalizadas pueden ser enviadas por administradores de la Dirección de Desarrollo Tecnológico e Informático (DDTI) de la universidad, desde el sistema web de gestión.
- Estas pueden dirigirse a todos los usuarios o a grupos específicos, como docentes o estudiantes de una facultad o carrera en particular.

Para el desarrollo del servicio de “Notificaciones y Eventos” se implementó un API Rest (Backend) utilizando el framework NestJs de Nodejs para manejar la lógica del servicio. La plataforma móvil está desarrollada en Flutter y se integró el servicio de notificaciones Firebase Cloud Messaging (FCM). Además, se utilizó el framework NextJs que usa librerías React para el desarrollo del sistema web destinada a la gestión de notificaciones personalizadas. Todo el proceso de desarrollo se guio en el marco de trabajo SCRUM para garantizar una gestión ágil y eficiente del proyecto, como se muestra en la **Figura 2**.

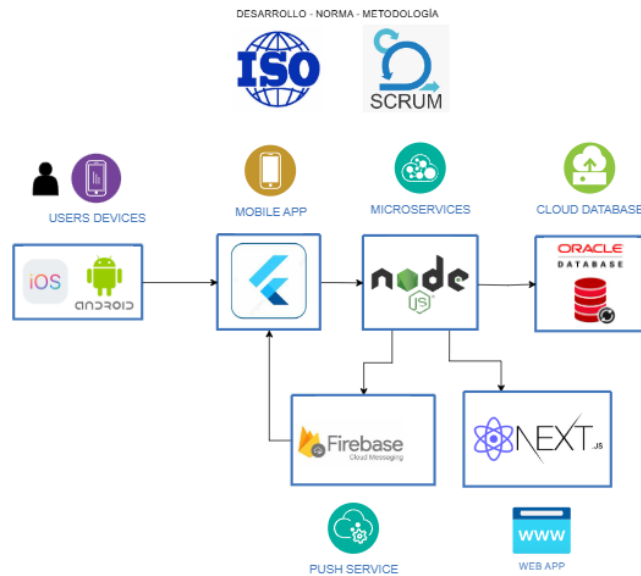


Figura 2 Arquitectura de la aplicación

Metodología

La técnica de investigación para el desarrollo del marco conceptual es del tipo documental, accediendo a las bases de datos científicas disponibles en el portal web de la UTN como IEEEExplore, ScienceDirect, entre otros motores de búsqueda especializadas en la investigación académica como Google Scholar. El instrumento de investigación a utilizar para la gestión de bibliografía es Mendeley.

Para el cumplimiento de los objetivos planteados, se llevarán a cabo las siguientes actividades:

Actividad 1: Se realizará una investigación de literatura científica y académica en relación con el desarrollo de aplicaciones móviles de carácter académico y la implementación del servicio de notificaciones en aplicaciones móviles.

Actividad 2: El desarrollo seguirá la metodología SCRUM y se dividirá el proceso en 3 partes principales:

- La fase de Pre-Juego, se definirán los requisitos del servicio en colaboración con los stakeholders. Se elaborará el backlog del producto y se planificarán las primeras iteraciones.
- La fase de Juego, se llevará a cabo el desarrollo del servicio, dividiendo el trabajo en iteraciones cortas llamadas "sprints". Se realizarán reuniones diarias de seguimiento y se ajustarán las tareas según sea necesario.

- La fase de Post-Juego, al finalizar cada sprint, se realizarán pruebas de aceptación para validar que las funcionalidades desarrolladas cumplan con los requisitos definidos en el backlog del producto.

Actividad 3: La evaluación del servicio de notificaciones de la plataforma móvil se guiará en los criterios específicos dentro de la norma ISO 25022, específicamente las características de eficacia, eficiencia y satisfacción. Para esta evaluación, se llevará a cabo una encuesta a los usuarios del producto de software sobre la experiencia de uso.

Justificación

El propósito de esta investigación resulta de interés pues se basa en la necesidad indispensable de proveer una herramienta ágil y de fácil acceso para la notificación de actividades y eventos universitarios. Los principales beneficiarios corresponden tanto al personal estudiantil como docente y administrativo que conforman la comunidad de la UTN de la ciudad de Ibarra.

Justificación tecnológica: Debido a la constante evolución de la tecnología y la notable integración de las aplicaciones móviles en la vida de las personas, el desarrollo e implementación de estas aplicaciones suponen un gran avance en cuanto al aprendizaje y los servicios que las instituciones académicas pueden ofrecer [3], [4]. Bajo este contexto, el desarrollo del servicio de notificaciones y eventos en la plataforma UTN-Móvil tiene un impacto positivo en la experiencia de los usuarios al recibir notificaciones, así como en su motivación para utilizar los servicios universitarios.

Justificación social: De igual manera, este trabajo fomenta el desarrollo de sistemas productivos inclusivos y sostenibles, así como la innovación, en consonancia con el Objetivo de Desarrollo Sostenible 9 de la ONU [5].

Trabajos Relacionados

Tabla 1 Trabajos relacionados

INVESTIGACIÓN	APORTE
----------------------	---------------

Contexto: Local

Implementación de una aplicación móvil para fortalecer la gestión de notificaciones e incidencias ciudadanas del GAD municipal de San Miguel de Ibarra, basado en la característica de adecuación funcional de la ISO/IEC 25010 [6].

Este estudio se enfoca en implementar una aplicación móvil para gestionar notificaciones e incidencias ciudadanas en San Miguel de Ibarra. Se aplica la norma ISO/IEC 25010 y utiliza SCRUM y Flutter. El punto diferencial del presente trabajo es la integración del servicio FCM con una arquitectura de microservicios, cumpliendo estándares de calidad de la norma ISO/IEC 25022.

Contexto: Nacional

Aplicación móvil usando el framework Flutter para el control de las tutorías académicas del sistema integrado de la Universidad Técnica de Ambato [7].

Este trabajo se enfoca en el diseño de una aplicación móvil para gestionar las tutorías académicas, utilizando el framework Flutter y la metodología Mobile-D. El aporte de este trabajo radica en la incorporación de la metodología SCRUM en el proceso de desarrollo del servicio de notificaciones.

Contexto: Internacional

Automated Educative Nudges to Reduce Missed Assignments in College [8].

Esta investigación emplea el framework Expo para el desarrollo de una aplicación móvil UI Boost para enviar notificaciones automáticas a los estudiantes en caso de que no hayan entregado tareas con fechas límite próximas. El presente trabajo de titulación integra el servicio de notificaciones FCM junto con el framework Flutter en la aplicación móvil.

Contexto: Internacional

Smart School System with Single ID based on RFID Through NFC using FCM Notification [9].

Esta investigación realiza una prueba piloto en una institución educativa de Yogyakarta integrando el servicio FCM para la notificación de préstamo de libros a estudiantes y padres de familia. El aporte del presente trabajo se enfoca en la

	implementación del servicio de notificaciones FCM para la oportuna comunicación dirigida a docentes y estudiantes de la universidad (usuarios de la aplicación).
Contexto: Internacional Android Application on e-notifier for College Events Management using Firebase [10].	En una conferencia internacional desarrollada en la India, se presentó una aplicación móvil Android para la gestión de eventos universitarios. Esta aplicación se conecta con Firebase para base de datos, autenticación y notificaciones. La contribución clave de este trabajo es la integración del servicio de notificaciones FCM en la plataforma UTN-Móvil para dispositivos Android y iOS.

I. MARCO TEÓRICO

1.1 Notificaciones y eventos en la Universidad Técnica del Norte

1.1.1 Proceso de notificaciones actual

Actualmente, debido a que la UTN no dispone de una aplicación móvil que ofrezca los diferentes servicios que provee la institución a su comunidad universitaria, la única forma de comunicación académica de manera inmediata es a través del correo electrónico institucional de la universidad vinculado con Microsoft. Esta vía de comunicación es utilizada principalmente por el personal administrativo de la UTN, como lo son las secretarías de carreras y el personal de los demás departamentos que maneja la institución como el departamento de Bienestar Universitario o el DDTI [11].

1.1.2 Sistema Informático Integrado Universitario

Además, existe otro canal de comunicación mediante el cual la universidad puede mantener informados a sus estudiantes y docentes. Este canal se basa en los portafolios independientes de cada uno, disponibles en el SIIU, al cual se puede acceder desde un navegador web [11].

Este medio informa principalmente sobre eventos y actividades académicas, como los horarios de clases, el registro académico de cada materia, las tareas y evaluaciones asignadas, los sílabos de las materias, eventos a desarrollarse en la universidad e información relacionada con procesos de matrícula, evaluación y becas [11]. Sin embargo, el principal problema de este canal de comunicación es que los usuarios deben iniciar sesión en el sistema web y buscar por su cuenta la información que necesiten en ese momento, lo cual resulta en una comunicación no inmediata. Es por ello la necesidad del desarrollo de un servicio de notificaciones de carácter inmediato para una comunidad más efectiva con la comunidad universitaria, con la implementación de una aplicación móvil para la UTN [11].

1.2 Introducción a las aplicaciones móviles

1.2.1 Evolución y tendencias en el desarrollo de aplicaciones móviles

Desde la introducción de las primeras generaciones de teléfonos inteligentes, como el primer smartphone basado en Android, el HTC Dream, en 2007, los cuales buscaban ofrecer ciertas características y aplicaciones básicas de un ordenador, en conjunto con pantallas táctiles y conectividad a Internet, se ha observado un incremento importante tanto en el desarrollo de dispositivos como de aplicaciones móviles [12], [13]. Hoy en día, se desarrollan aplicaciones móviles que satisfacen todo tipo de necesidades, incluyendo servicios bancarios, educativos, de comida, entretenimiento, entre otros[12].

Este desarrollo de aplicaciones para dispositivos móviles subyace en dos categorías generales: en primer lugar, el desarrollo de aplicaciones web con capacidad de respuesta para pantallas de teléfonos, tabletas y dispositivos similares, ejecutadas desde un navegador; y, en segundo lugar, el desarrollo de aplicaciones específicamente para dispositivos móviles [12]. Esto ha permitido que tanto el uso de las generaciones pasadas de PDAs (Asistentes Digitales Personales) como de los smartphones actuales se hayan promovido en diferentes áreas de la humanidad, como la educación, la medicina, los automóviles, entre otras [13].

Es por esto por lo que, en la última década, se ha evidenciado un crecimiento exponencial en cuanto al uso de dispositivos móviles. Este crecimiento se puede ver reflejado en el incremento de descargas de aplicaciones móviles de manera anual. En 2019, las principales tiendas de mercado de aplicaciones móviles como lo son Google Play para dispositivos Android y App Store para

dispositivos iOS, alcanzó un total de 2,7 millones de aplicaciones desplegadas. Adicionalmente, según Statista en 2022, se registró un total de 230 mil millones de descargas de aplicaciones móviles, comprendiendo un total de 3,5 millones de aplicaciones en Google Play y 1,64 millones en App Store [2], [12].

Este fenómeno se refleja igualmente en América Latina, donde hasta 2018 la adopción de teléfonos inteligentes llegó al 39,1 %, mostrando una tendencia al alza a medida que los servicios digitales se trasladan a plataformas móviles [14]. En el contexto de Ecuador, se estimó que en 2018 el 57 % de la población contaba con un teléfono inteligente, y se proyecta que para el año 2025 aproximadamente el 73 % de los habitantes adoptará esta tecnología [14].

De igual forma, con el paso de los años, las personas continúan adquiriendo nuevas generaciones de teléfonos inteligentes y requieren cada vez mejores características y más servicios para cumplir sus necesidades [12]. Debido a esto, resulta esencial para los desarrolladores de aplicaciones móviles aprovechar al máximo la capacidad de estas nuevas generaciones tanto en hardware como en software e implementar nuevas herramientas y servicios en su desarrollo para la satisfacción de los usuarios finales.

1.2.2 Importancia de las notificaciones en aplicaciones móviles

Las notificaciones surgen como un servicio estratégico fundamental en el uso de dispositivos móviles, permitiendo a los usuarios de dichos dispositivos obtener información relevante de sus aplicaciones de uso diario [15]. Con la implementación de un servicio de notificaciones en una aplicación móvil, los usuarios de dicha aplicación pueden recibir mensajes cortos en las pantallas de sus dispositivos móviles lo cual permite una comunicación rápida y efectiva. Esto, garantiza una mayor probabilidad de mantener más usuarios activos mediante la interacción y redirección de los usuarios hacia la aplicación móvil de destino [16].

Además, de la principal ventaja que suponen las notificaciones de mantener activos a los usuarios de una aplicación, estas también permiten aumentar el número de transacciones y efectivización de procesos realizados en dicha aplicación [16].

1.3 Arquitectura de microservicios

En el área de la ingeniería de software, existen dos principales arquitecturas entorno al desarrollo de sistemas informáticos; la arquitectura monolítica y la arquitectura de microservicios [17]. Esta última, aumentando su popularidad alrededor del año 2012 [17].

1.3.1 Conceptos y beneficios de la arquitectura de microservicios

En primera instancia, es necesario conocer el enfoque tradicional en el desarrollo de sistemas. La arquitectura monolítica se basa principalmente en el desarrollo de aplicaciones que se crean bajo una sola base, en la cual se implementan múltiples servicios que no son independientes y se comunican entre sí, fusionando la parte del cliente con la lógica del negocio y del servidor [17], [18].

Con la evolución de los sistemas de software, los desafíos se incrementan y complican la agilidad del desarrollo. Es por ello por lo que surge la arquitectura basada en microservicios para facilitar la construcción de aplicaciones más complejas con una integración optimizada [19].

La arquitectura de microservicios en el desarrollo de software es un enfoque arquitectural que se refiere a la construcción de software como un conjunto de servicios desplegados de forma independiente y autónoma [18][20]. Este enfoque tiene como objetivo principal, la separación y organización de sistemas informáticos entre módulos, considerando la lógica del negocio y manejando una descentralización del lenguaje y datos [17], [18].

La transición de un enfoque monolítico a uno de microservicios asegura ciertos beneficios con respecto a la escalabilidad, agilidad y mantenimiento a futuro del sistema, siempre y cuando se identifique de manera correcta los límites y funciones de los microservicios [19].

Otra ventaja del desarrollo de microservicios es que, al estar cada servicio enfocado únicamente en llevar a cabo una funcionalidad del sistema, se permite mantener un desarrollo mucho más rápido [18]. Además, este desarrollo ágil permite al equipo de trabajo implementar continuamente nuevas funcionalidades sin afectar la estabilidad general del sistema ni del resto de sus componentes [18].

1.3.2 Implementación de microservicios en aplicaciones móviles

Con el avance en el desarrollo de sistemas de información, desde plataformas orientadas a equipos de escritorio hasta las aplicaciones móviles actuales, el mantenimiento de sistemas y aplicaciones se ha vuelto más recurrente [21]. En especial, las aplicaciones móviles consumen

muchos servicios de aplicaciones y de API's para un manejo de los datos de manera indirecta, contrario a las aplicaciones monolíticas [21].

Debido a esto, el desarrollo de aplicaciones mediante la construcción de componentes independientes y distribuidos resulta en un medio eficaz para la universalidad e interoperabilidad de varios sistemas [21], [22].

1.4 Metodología ágil Scrum

Scrum es un marco de trabajo ágil que se guía en un enfoque colaborativo hacia personas, grupos de trabajo y organizaciones siguiendo los valores fundamentales del Manifiesto Ágil [23], [24]. El principal objetivo de trabajar con la metodología SCRUM es la entrega de soluciones que generen valor a través de todo el ciclo de vida que comprende su ejecución [23].

1.4.1 Principios y prácticas de Scrum

Scrum se fundamenta en el empirismo y el pensamiento Lean, priorizando la experiencia y la observación para la toma de decisiones y minimizando el desperdicio para centrarse en lo esencial [23]. Además, se basa en una metodología iterativa e incremental para la optimización de resultados durante el desarrollo mediante la gestión de riesgos, involucrando a grupos de trabajo que comparten las habilidades necesarias [23].

Asimismo, Scrum ha probado ser sumamente eficaz en ámbitos industriales, donde el enfoque ágil en el desarrollo de software se ha establecido como una práctica que favorece la entrega constante de valor. Esta metodología ofrece una mejor capacidad de adaptación ante escenarios cambiantes, promueve la comunicación activa de los clientes y eleva sus niveles de satisfacción [25].

El marco de trabajo Scrum se complementa con la aplicación de cinco valores fundamentales que orientan el trabajo, las acciones y el comportamiento de un equipo, con el objetivo de entregar soluciones de calidad, partiendo de una correcta planificación [23], [24]. Estos valores son: compromiso, foco, franqueza, respeto y coraje [23].

1.4.2 Roles de SCRUM

El Scrum Team es la base de todo el marco de trabajo Scrum y está compuesto por 3 roles principales [23]. Este equipo tiene como objetivo principal alcanzar los objetivos del proyecto mediante el desarrollo de funciones específicas de cada rol en cada fase del proyecto, también denominada Sprint [23]. Los 3 roles principales son:

- **Scrum Master:** se centra en guiar al equipo Scrum para que implemente Scrum de la manera más efectiva y eficiente posible. El Scrum Master elimina los obstáculos que puedan impedir el progreso del equipo [23].
- **Product Owner:** se enfoca en definir los objetivos del producto, gestionar el Product Backlog y priorizar sus elementos. El Product Owner busca maximizar el valor del producto y comunica claramente los requisitos del negocio al equipo Scrum [23].
- **Developers:** se enfocan en crear incrementos al producto al final de cada Sprint. Los Developers poseen todas las habilidades necesarias para generar valor en cada Sprint y colaboran para desarrollar los elementos del Sprint Backlog seleccionados durante la planificación [23].

1.4.3 Fases de Scrum

Dentro de Scrum, se desarrollan una serie de eventos en los cuales se presenta una oportunidad para revisar y mejorar los artefactos de Scrum, que pueden ser: el Product Backlog, el Sprint Backlog y el Increment [23]. A continuación, en la **Figura 3**, se presenta cada evento y artefacto de valor en la metodología Scrum.

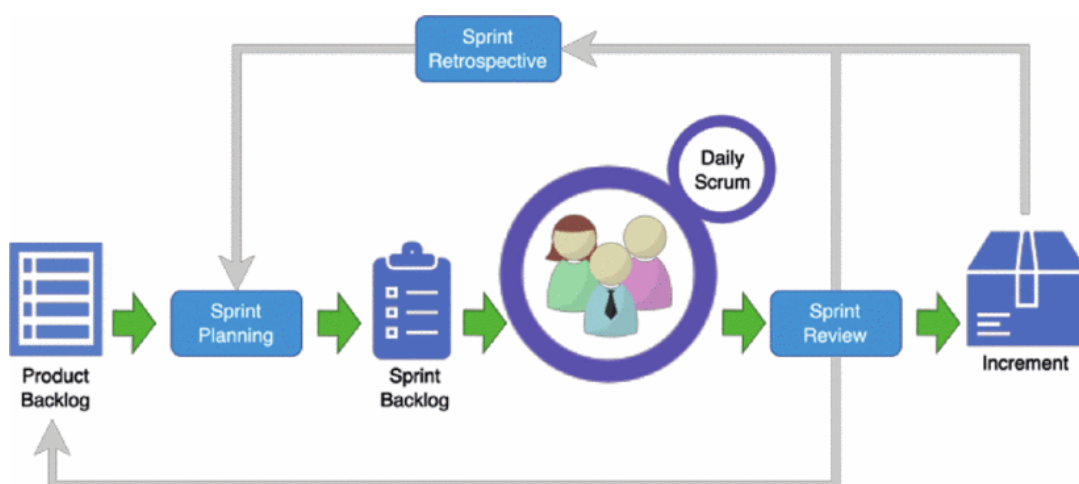


Figura 3 Procesos de Scrum [26]

- **Sprint:** Período de tiempo fijo (máximo 1 mes), durante el cual se crea un Increment al producto [23].

- **Product Planning:** Definición de los objetivos y creación de la lista de actividades (Product Backlog) a desarrollarse para el cumplimiento del producto final [27].
- **Sprint Planning:** Selección de las actividades del Product Backlog que se llevarán a cabo durante el presente Sprint [27].
- **Sprint Backlog:** Lista de trabajo o actividades a desarrollarse durante el Sprint [26].
- **Daily Scrum:** Reuniones cortas (15 minutos) con la presencia del Scrum Master y los Developers para evaluar el rendimiento del equipo de trabajo en relación a los objetivos planteados [27].
- **Sprint Review:** Reunión que se desarrolla cada mes con el propósito de revisar el trabajo completado y el cumplimiento de los objetivos [27].
- **Sprint Retrospective:** Análisis del trabajo realizado e identificación de posibles mejoras para el desarrollo del próximo Sprint [27].
- **Increment:** Resultado o funcionalidad nueva desarrollada por el equipo Scrum durante cada Sprint, es decir, los avances para el desarrollo total del producto [23].

1.4.4 Personal Process Software (PSP)

En conjunto con la metodología Scrum, PSP se enfoca en la mejora de la construcción de software de cada individuo involucrado en el desarrollo del producto [28].

PSP se basa en principios de planificación y calidad, enfatizando que cada desarrollador debe planificar su trabajo según su información personal y medir sus procesos para mejorar continuamente [28]. A continuación, se presentan sus fases en conjunto con la metodología Scrum.

- **Fase de pre-juego:** La fase de pre-juego en la metodología de Scrum corresponde a la fase de preparación y definición del proyecto, en el cual se obtienen las necesidades del cliente y se crea el Product-Backlog [28].
- **Fase de juego:** La fase de juego corresponde a la fase de desarrollo del proyecto, donde se seleccionan las actividades del Sprint Backlog a desarrollarse, en conjunto con las prácticas de PSP [28].
- **Fase de post-juego:** El post-juego corresponde a la etapa de entrega, en la cual se integran los nuevos Increment y se realizan pruebas de evaluación y de verificación de funcionalidad [28].

1.5 Normas de evaluación de calidad

En la actualidad, es muy importante garantizar la calidad en el desarrollo de un producto de software, puesto que, es la calidad la que determina el tiempo de vida útil del producto [29].

Evaluar la calidad de un producto de software requiere analizar tanto el producto terminado como los procesos empleados en su creación, usando una metodología que permita medir tanto los atributos internos como externos, asegurando así que se satisfagan las necesidades y requisitos del cliente [30].

Dentro de las series de normas que ayudan a los desarrolladores a medir y evaluar la calidad de un producto, se encuentra la familia ISO/IEC 25000, también denominadas Requisitos y Evaluación de la Calidad de Sistemas y Software (SQuare) [29]. Esta familia de normas se organiza de la siguiente manera, como se aprecia en la **Figura 4**.

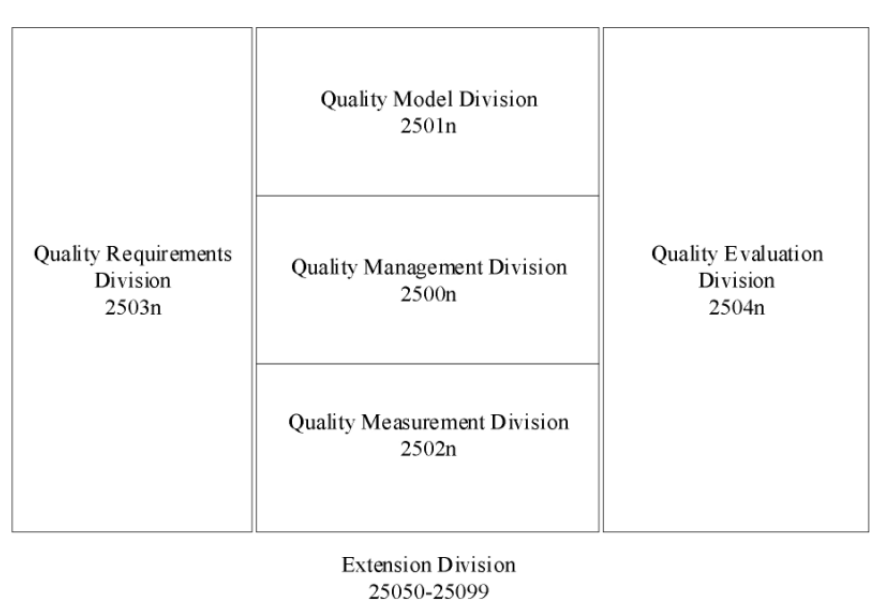


Figura 4 Organización de la familia de normas ISO/IEC 25000

- **División de Gestión de calidad:** Está constituida por la ISO/IEC 25000, que define la arquitectura y terminología, y la norma ISO/IEC 25001, que ofrece directrices para la planificación y gestión de la evaluación y especificaciones del software [30].
- **División de Modelos de calidad:** Incluye la norma ISO/IEC 25010, que define ocho características para evaluar el software: adecuación funcional, eficiencia de desempeño, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y

portabilidad. Además, incorpora la norma ISO/IEC 25012, que ofrece métricas para medir la calidad de los datos [30].

- **División de Medición de calidad:** Contiene varias normas que detallan las métricas necesarias para medir y la manera de medir la calidad del software como la norma ISO/IEC 25020, 25021, 25022, 25023 y 25024 [30].
- **División de Requisitos de calidad:** Esta sección está constituida por la norma ISO/IEC 25030, que clarifica los requisitos de calidad del software para ser utilizados en el proceso de evaluación [30].
- **División de Evaluación de calidad:** Incluye la norma ISO/IEC 25040 que establece un marco para evaluar software, abordando entradas, restricciones y recursos necesarios, la ISO/IEC 25041, que proporciona directrices para desarrolladores, compradores y evaluadores sobre cómo realizar evaluaciones de software y la ISO/IEC 25042 y 25045, que detallan módulos específicos para evaluar la estructura y la recuperabilidad del software, respectivamente [30].

1.5.1 Norma ISO/IEC 25022: Modelos de calidad en uso de sistemas informáticos

La norma ISO/IEC 25022 forma parte de la familia de normas Square y establece las medidas de calidad en uso para evaluar las características especificadas en la norma ISO/IEC 25010 [31]. Esta norma, tiene como objetivo principal satisfacer las necesidades de los clientes o usuarios en relación con la calidad del producto final [31].

Dentro de la norma, se presentan varios criterios para medir la calidad en uso de un producto. Entre estos criterios, se destacan: la efectividad, que se refiere a la precisión y la integridad con las que los usuarios alcanzan sus objetivos específicos; la eficiencia, que mide la relación entre los recursos utilizados y la consecución de dichos objetivos; y la satisfacción del usuario, que evalúa en qué medida se satisfacen las necesidades del usuario en contextos específicos de uso [30], [32].

1.5.2 Evaluación de la calidad en uso de servicios en aplicaciones móviles

Dado el crucial papel de garantizar la calidad en los productos de software y el notable crecimiento en el desarrollo de aplicaciones móviles mencionado en capítulos anteriores, evaluar la calidad externa en este contexto ofrece numerosas ventajas, las cuales incluyen mejorar la eficiencia y la portabilidad del producto final, así como identificar incidentes y

áreas de mejora en el software, facilitando su corrección y promoviendo una mejora continua en la aplicación [33].

1.6 Experimento para la selección del lenguaje de programación

Para el presente trabajo se realizó una Revisión Sistemática de la Literatura (SLR) con el objetivo de identificar los lenguajes de programación y frameworks más adecuados para el desarrollo de APIs REST, las cuales consisten en una serie de interfaces con operaciones HTTP que siguen una estructura organizada y escalable, en el contexto del backend [34][35]. Para ello, se aplicaron las metodologías propuestas por Kitchenham y PRISMA [36], lo cual permitió establecer un proceso riguroso de selección, clasificación y análisis de publicaciones científicas [37].

En la **Figura 5**, se describe el porcentaje de evaluaciones realizadas en los lenguajes de programación a través del tiempo.

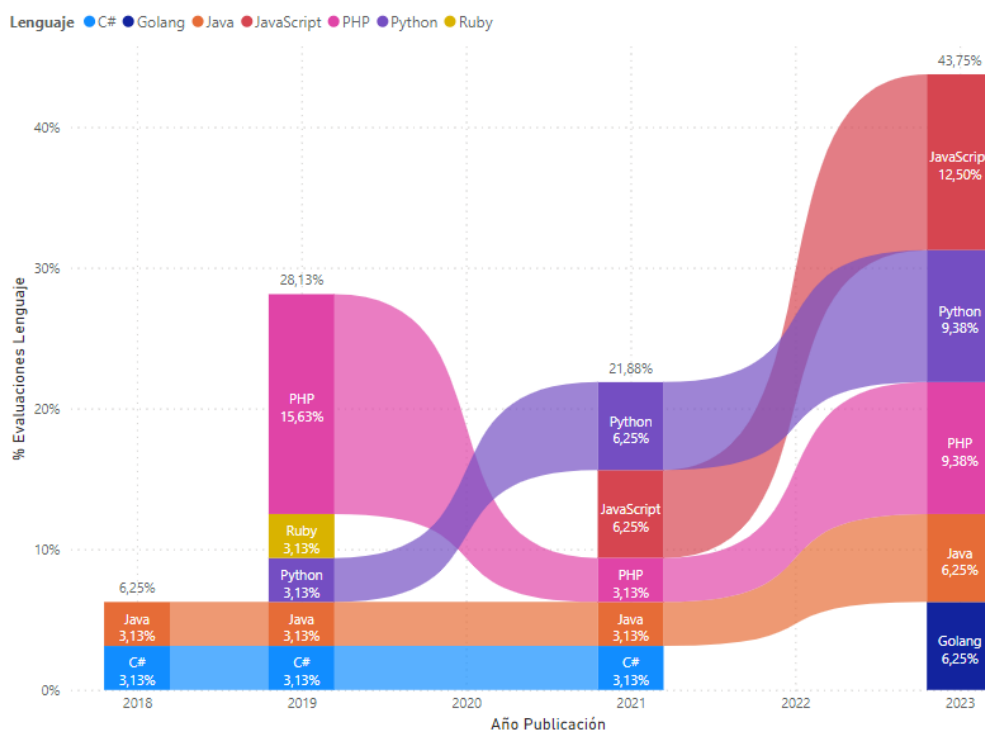


Figura 5 Porcentaje de evaluación de lenguajes

Fuente: Equipo UTN móvil

Como resultado del SLR se identificaron cuatro tecnologías candidatas, destacadas por su frecuencia de uso y el rendimiento reportado, de acuerdo con lo mostrado en la **Tabla 2**.

Tabla 2 Tecnologías seleccionadas para el experimento

LENGUAJE DE PROGRAMACIÓN	FRAMEWORK
Javascript/typescript	NestJS

Python	Flask
Java	Spring Boot
SQL	Oracle Apex

En el desarrollo de un experimento computacional, el objetivo principal consiste en analizar el comportamiento de los factores de interés bajo condiciones controladas y reproducibles, siguiendo principios metodológicos que permitan evaluar diferentes parámetros de manera precisa como el rendimiento de un sistema [38].

Una vez seleccionadas las tecnologías, se llevó a cabo el experimento comparativo en el que se desarrolló una API REST utilizando cada una de las tecnologías seleccionadas. La evaluación de estas implementaciones se estructuró en torno a dos variables clave de complejidad:

- Profundidad de consultas (1-4 niveles de JOIN)
- Volumen de registros (1, 10, 100, 1,000, 10,000 y 100,000 registros)

Esto permitió simular distintos escenarios de carga y complejidad, aproximándose a un contexto que puede enfrentar una aplicación en producción. Las métricas observadas incluyeron el tiempo de respuesta, uso de CPU y uso de memoria.

Como resultado, a pesar de que todos los frameworks demostraron ser funcionales, NestJS (basado en Node.js) presentó el mejor rendimiento en términos de tiempo de respuesta bajo carga, además de una curva de aprendizaje más favorable, lo que lo convirtió en la opción más equilibrada entre rendimiento y facilidad de desarrollo.

1.7 Herramientas de desarrollo

1.7.1 Node.JS

Node.JS es un entorno de ejecución basado C++ [39], [40], el cual permite el desarrollo en los lenguajes de Javascript y Typescript [41]. Además, Node.JS emplea un esquema asíncrono y bucles de eventos que mejora el rendimiento del servidor [41], [42].

Este entorno de ejecución es comúnmente utilizado para el desarrollo backend de aplicaciones y dispone de sus propios frameworks en este contexto como ExpressJS y NestJS [39], [42].

1.7.1.1 Nest.JS

Nest.JS es un framework basado principalmente en el lenguaje de programación Typescript [41]. Este framework integra varios paradigmas de programación que permiten un

desarrollo más estructurado, entre ellos resaltan la programación orientada a objetos, inyección de dependencias y detección de errores antes y durante su ejecución gracias al uso de Typescript [41], [42]. Nest está estructurado de la siguiente manera:

- **Controlador:** Gestión de solicitudes y respuestas HTTP al cliente [41].
- **Proveedor/Servicio:** Se encarga de solicitudes externas como microservicios y conexión a base de datos [41].
- **Middleware:** Función que se ejecuta antes del controlador de ruta, con acceso a las solicitudes y respuestas [41].

1.7.2 Oracle Database

Oracle Database 19c es uno de los sistemas comerciales de gestión de bases de datos relacional más rápidos y robustas en términos de importación de datos, es por ello por lo que esta es una de las opciones de preferencia para grandes empresas que manejan mucha información [43]. Entre sus principales características destacan:

- **Alta eficiencia:** Mantiene una buena eficiencia en términos de velocidad y rendimiento en comparación con otros sistemas de gestión de bases de datos como MySQL y SQL Server [43].
- **Funcionalidades avanzadas:** Proporciona una amplia variedad de funciones para la gestión de bases de datos, lo que lo convierte en una opción sólida para entornos empresariales [43].
- **Soporte para transacciones OLTP:** Está optimizado para el procesamiento de transacciones en línea, lo cual lo convierte en una opción idónea para aplicaciones que requieren alto rendimiento en operaciones de lectura y escritura [43].

1.7.2.1 PL/SQL

PL/SQL es un lenguaje de programación desarrollado por Oracle, inspirado en Ada, que es ampliamente utilizado por los desarrolladores y administradores de bases de datos [44].

Este lenguaje de programación se integra con el lenguaje general de bases de datos SQL, permitiendo ejecutar de manera directa sentencias SQL en funciones y procedimientos PL/SQL y viceversa [44]. A continuación, se presentan los principales usos de este lenguaje de programación.

- Aplicación de procedimientos almacenados y triggers para el desarrollo de la lógica del negocio de nuestro sistema.
- Generación y gestión de documentos XML.
- Automatización de tareas administrativas de bases de datos, incluyendo la seguridad a nivel de filas y la gestión de segmentos de reversión.

[44]

1.7.3 Next.js en el desarrollo de aplicaciones web

Next.js es un framework de desarrollo web basado en React, cuya función principal es su renderizado del lado del servidor [45]. Sin embargo, para entender de mejor manera todas las ventajas que nos proporciona este framework, primero debemos entender sus bases [45].

Document Object Model (DOM)

En el contexto de las aplicaciones web, para que se muestre una página web se realiza una solicitud al navegador desde el que se quiere acceder, el cual se encarga de la renderización de todo el documento a manera de árbol con etiquetas HTML. A esta estructura se la denomina DOM [45].

React

React es una librería desarrollada en el lenguaje de programación de JavaScript, usada principalmente para la creación interfaces de usuario interactivas con componentes altamente reutilizables en todo el sistema [45]. Esta librería hace uso de un DOM virtual, realizando cambios en memoria antes de actualizar el DOM real para mejor rendimiento en la renderización de las páginas y componentes web [45].

Client Side Rendering (CSR)

React utiliza la renderización del lado del cliente, donde el contenido es generado por JavaScript en el navegador. Aunque la carga inicial es más lenta, las siguientes cargas de página son más rápidas [45].

Server Side Rendering (SSR)

Por otro lado, Next.js usa el renderizado del lado del servidor, enviando una respuesta HTML desde el servidor al navegador, que luego renderiza la página, en milisegundos [45].

En este contexto, Next.js se muestra como una buena opción para el desarrollo de aplicaciones web, garantizando las siguientes ventajas:

- **Optimización para Motores de Búsqueda (SEO):** Facilita la creación de títulos y palabras clave para cada página, mejorando el SEO mediante el uso de componentes Head [45].
- **CSS Integrado:** Soporta la importación de estilos CSS desde archivos JavaScript, acelerando la renderización [45].
- **Carga Diferida:** Permite importar componentes dinámicamente durante el tiempo de ejecución, mejorando el rendimiento inicial al reducir la cantidad de JavaScript necesario [45].
- **División Automática de Código:** Separa el paquete de la aplicación en partes más pequeñas necesarias para cada punto de entrada, reduciendo el tiempo de carga inicial [45].

1.7.4 Flutter en el desarrollo de aplicaciones móviles

Flutter es definido como un kit de desarrollo de software (SDK) orientado al desarrollo de aplicaciones móviles, el cual fue lanzado por Google en 2017 [46][47]. Una de las principales características de este SDK móvil, es su desarrollo multiplataforma de código abierto, el cual está basado en el lenguaje de programación Dart [47].

Lenguaje de programación Dart

De igual manera que con Flutter, Dart fue desarrollado por Google. Además, este es un lenguaje orientado a objetos tipado muy intuitivo con una curva de aprendizaje relativamente baja [46].

Por otro lado, Dart permite dos tipos de compilación, compilación anticipada (AOT), la cual otorga un mayor rendimiento al convertir el código a uno nativo de máquina y la compilación al momento (JIT) que permite una recarga de la aplicación sobre la marcha, agilizando el desarrollo [46].

1.8 Notificaciones Push

Las notificaciones push, se encargan de mostrar la información necesaria y relevante en un *pop-up* en la pantalla de los usuarios de cualquier aplicación ya sea móvil o web [16]. Este tipo de notificaciones, son identificadas también como un servicio en la nube, notificaciones de alerta como servicio (ANS) [48].

1.8.1 Funcionamiento y tipo de notificaciones push

Para el envío de notificaciones push, se lo puede realizar a través de varios protocolos como HTTP, web sockets, SSE, OMA-PUSH, RFC3428, APNs para el envío de notificaciones en dispositivos iOS y FCM para dispositivos Android [48], [49]. Las notificaciones de aplicaciones ya sean web o móvil se pueden segmentar de la siguiente manera:

- **Notificaciones in-app:** son los mensajes que se muestran dentro de la aplicación, que no necesariamente se muestran como una ventana emergente al igual que las notificaciones push tradicionales y que suelen reflejar un cambio en el ícono de la aplicación como un ícono o el número de notificaciones [50].
- **Notificaciones push basadas en actividades:** son los mensajes generados por la aplicación resultantes de un evento o acción específica del usuario [50].

1.8.2 Buenas prácticas en el envío de notificaciones

En el ámbito de las notificaciones, existen algunas estrategias que nos aseguran la permanencia y actividad de los usuarios en las aplicaciones móviles como [16]:

- **Personalización:** notificaciones personalizadas a cada usuario con la finalidad de que estos completen tareas o transacciones dentro de la aplicación [16].
- **Optimización:** entrega de notificaciones en tiempo real de manera oportuna sobre únicamente actividades relevantes para el usuario [50][51].
- **Inhabilitación:** permite a los usuarios de la aplicación inhabilitar la recepción de notificaciones en caso de experimentarlas intrusivas [50][51].

1.8.3 Firebase Cloud Messaging

FCM es un servicio multiplataforma de notificaciones de Google [52]. Existen varias maneras gestionar y segmentar el envío de notificaciones a los dispositivos de una aplicación, algunas de estas son:

- Envío de notificaciones global (todos los dispositivos registrados de la aplicación) [53].
- Envío de notificaciones a un usuario en específico [53].
- Envío de notificaciones a un tema, al cual están suscritos un grupo de dispositivos [53].

De igual manera, podemos conectar nuestra aplicación cliente con FCM a través de su SDK o de un API [52]. A continuación, en la **Figura 6**, se muestra la arquitectura del servicio FCM.

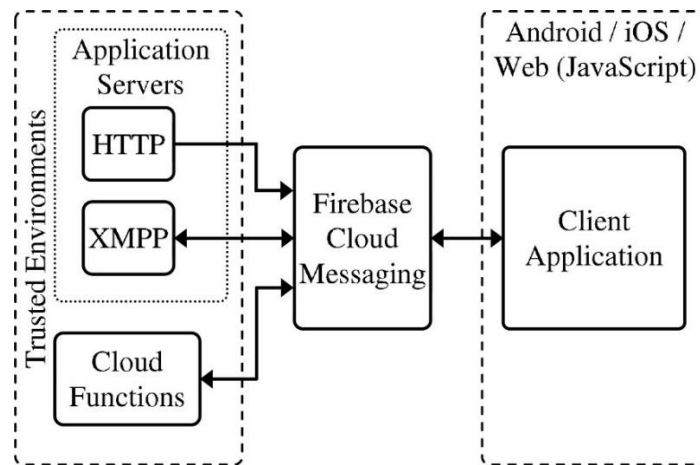


Figura 6. Arquitectura de FCM [52]

1.9 Desarrollo de aplicaciones multiplataforma

Las aplicaciones multiplataforma son aquellas que se pueden ejecutar desde un mismo código de programación compartido para más de una plataforma como Android, iOS y web, siendo el SDK Flutter uno de los más populares actualmente [54].

1.9.1 Ventajas y desafíos de aplicaciones multiplataforma

Las principales ventajas referentes a las aplicaciones multiplataforma tienen que ver con el esfuerzo dedicado de los desarrolladores de la aplicación, puesto que al ser una menor cantidad de código supone un ahorro en el tiempo y una simplificación de recursos durante la fase de desarrollo y mantenimiento principalmente [55].

Sin embargo, a pesar de todas las ventajas que lleva consigo el desarrollo de aplicaciones multiplataforma se puede incurrir en algunas desventajas principalmente en temas de aprovechamiento de recursos, y demás características avanzadas que ofrece un desarrollo nativo [55].

1.10 Trabajos Relacionados

En el estudio realizado por R. Pérez, se desarrolló una aplicación móvil para la gestión y notificación de incidencias del municipio de la ciudad de Ibarra [6]. Para la elaboración de esta aplicación, se menciona el marco de desarrollo móvil Flutter, siguiendo el marco de desarrollo SCRUM. Para la medición de los resultados, el estudio menciona la presentación

de una encuesta a 100 ciudadanos de Ibarra, en el cual se demuestra un índice de favorabilidad del 97.33% en cuanto a la calidad del sistema y un índice del 92.75% en cuanto a la satisfacción del usuario con la aplicación desarrollada. Sin embargo, una limitante del estudio es que no implementa un servicio de notificaciones push para la aplicación móvil.

En el diseño de una aplicación móvil para la Universidad Técnica de Ambato, P. Bautista desarrolló una aplicación móvil para la gestión de tutorías académicas [7]. Para su diseño, el estudio menciona la metodología Mobile-D, ASP.NET CORE 5 y Flutter. Antes de iniciar el desarrollo, realizó una encuesta sobre la necesidad de una aplicación móvil para gestionar las tutorías impartidas por los docentes, resultando en que el 94.8% de los docentes consideraron necesario este desarrollo. Este trabajo se enfoca principalmente en la notificación automática sobre las tutorías tanto para docentes como para estudiantes. Por lo tanto, el aporte de mi trabajo radica en el desarrollo de un servicio para notificaciones automáticas y para la notificación de eventos personalizados, utilizando una arquitectura basada en microservicios para la aplicación móvil.

En una institución educativa de Yogyakarta, Rokhmadi et al., llevaron a cabo una prueba piloto de un sistema escolar inteligente desarrollado para la gestión académica en una escuela [9]. En el sistema implementaron el servicio de notificaciones de Firebase FCM y tecnología RDIF y NFC, con el objetivo de informar a los dispositivos móviles de los estudiantes y padres de familia sobre el préstamo de libros, asistencia, entre otros. De igual manera, llevaron a cabo una encuesta a 55 estudiantes con el objetivo de evaluar la efectividad del sistema, mostrando que un 25.15% de los estudiantes estuvieron mayormente de acuerdo con el uso de la aplicación y un 55.15% estuvieron de acuerdo. Este trabajo utiliza una base de datos MySQL para el sistema de gestión escolar, lo cual puede limitar la escalabilidad y el rendimiento del sistema si la cantidad de usuarios y datos aumenta significativamente en el futuro.

En la Conferencia Internacional sobre Tecnología Aeroespacial y Comunicación en Coimbatore, K. S. Lakshmi et al., presentaron el desarrollo de una aplicación móvil Android para la gestión de eventos de una universidad [10]. La aplicación se centró en la implementación del servicio de notificaciones para todos los eventos de la universidad de manera categorizada. Esta aplicación móvil se desarrolló con Android Studio y para la gestión del backend de la aplicación se utilizaron los servicios de Firebase. Como resultado, la aplicación proporcionó una manera eficiente para la notificación de eventos universitarios fomentando la participación de sus estudiantes. Sin embargo, algunas limitaciones del estudio

incluyen que la aplicación no es multiplataforma, lo que restringe el acceso a personas que no usan dispositivos Android. Además, solo gestiona notificaciones personalizadas, sin implementar notificaciones push automáticas para diferentes procesos académicos.

En la Universidad Tecnológica de Israel, N. Jimenez, implementó un servicio de notificaciones académicas específicamente para el módulo de estudiantes del sistema de la universidad ya tiene integrado [56]. En este trabajo se menciona el uso de las tecnologías de Xamarin, para el desarrollo móvil en conjunto con el servicio de Google FCM para el envío de notificaciones push. Para la evaluación del sistema móvil la investigación menciona la aplicación de una encuesta dirigida a los estudiantes con métricas como facilidad de uso y personalización. El aporte que supone mi trabajo supone el uso de la norma ISO/IEC 25022 para evaluar la calidad en uso del software, concretamente el servicio de notificaciones y eventos para todos usuarios de la aplicación móvil.

II. DESARROLLO

El presente proyecto tiene como objetivo desarrollar un servicio backend para la gestión de notificaciones y eventos, acompañado de una aplicación web que facilite su administración, utilizando el marco de trabajo SCRUM para garantizar una ejecución ágil y eficiente.

2.1 Fase de Pre-juego

2.1.1 Equipo Scrum

El grupo de trabajo involucrado para el desarrollo del servicio de notificaciones en este proyecto se detalla en la **Tabla 3**.

Tabla 3 Roles Scrum del proyecto

Nombre	Rol	Cargo
PhD. Cathy Guevara	Product Owner	Docente de la UTN y directora del presente trabajo de integración curricular.
PhD. Antonio Quiña	Stakeholder	Docente de la UTN y director del proyecto plataforma UTN-móvil.

DDTI	Stakeholder	Director de la Dirección de Desarrollo Tecnológico Informático (DDTI) de la UTN.
Leonardo Flores	Scrum Master Equipo de desarrollo	Tesista.

2.1.2 Levantamiento de requisitos

Durante la fase de planificación inicial del proyecto, se realizó el levantamiento de requisitos en conjunto con el equipo Scrum. Conforme a este marco de trabajo, se definieron las historias de usuario que describen las principales funcionalidades a desarrollar, alineadas con los objetivos y el alcance establecidos para el proyecto.

En la **Tabla 4**, se presenta la planificación del proyecto siguiendo el marco de trabajo Scrum.

Tabla 4 Planificación Sprints

Sprint	Actividad	Duración
Sprint 0	Elicitar los requisitos de software. Definir las historias de usuario. Diseñar la arquitectura del software. Configurar el entorno de trabajo.	1 semana
Sprint 1	Desarrollo del API (Backend) para el envío de notificaciones push	3 semanas
Sprint 2	Desarrollo del API para las notificaciones basadas en eventos académicos correspondientes a docentes y estudiantes.	3 semanas

	Desarrollo del cliente móvil para la visualización de las notificaciones académicas de los usuarios docentes y estudiantes.	
Sprint 3	Desarrollo del cliente web para la gestión de notificaciones.	3 semanas
Sprint 4	Desarrollo de funcionalidades pendientes y pruebas funcionales	2 semana

2.1.2.1 Encuestas diagnósticas para el levantamiento de requerimientos funcionales

Objetivo

El objetivo de estas encuestas fue identificar las funcionalidades más utilizadas dentro del portafolio académico actual, así como determinar qué tipos de notificaciones académicas serían consideradas útiles para optimizar la organización y comunicación tanto de los docentes como de los estudiantes.

Metodología Aplicada

Para identificar los tipos de notificaciones académicas más relevantes desde perspectiva de los usuarios del sistema, se aplicaron dos encuestas diferenciadas dirigidas a docentes y estudiantes de la UTN.

Ambas encuestas fueron de tipo estructurado, conformadas principalmente por preguntas cerradas y con escala de Likert, con el objetivo de cuantificar de forma objetiva las percepciones y preferencias de los participantes. Los cuestionarios fueron elaborados y distribuidos a través de la plataforma Microsoft Forms.

En cuanto al perfil de los encuestados, se incluyeron docentes y estudiantes de todas las facultades académicas de la universidad. No se empleó un muestreo estratificado ni se

aplicó una distribución proporcional por facultad, por lo que la participación varió entre unidades académicas según la disponibilidad y voluntad de los participantes.

La encuesta dirigida a docentes registró un total de 118 respuestas válidas, mientras que la correspondiente a estudiantes obtuvo 405 respuestas. Estos resultados constituyeron una base sólida para fundamentar la selección de los tipos de notificaciones a implementar en el sistema.

En la **Figura 7** se muestra la distribución de docentes encuestados por facultad.

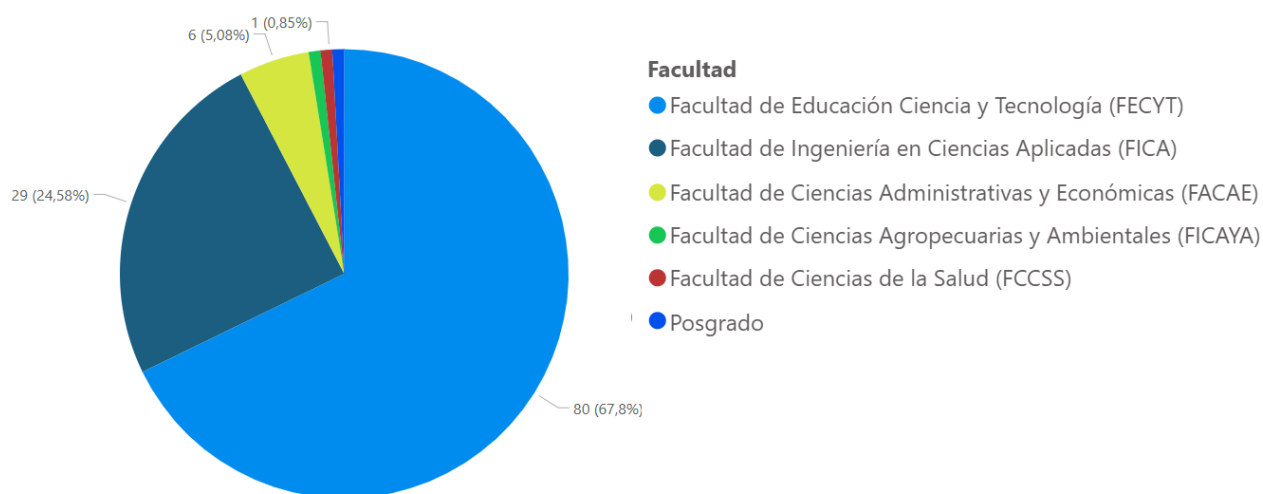


Figura 7 Distribución docentes- Encuesta diagnóstica

Fuente: Propia

En la **Figura 8** se muestra la distribución de estudiantes encuestados por facultad.

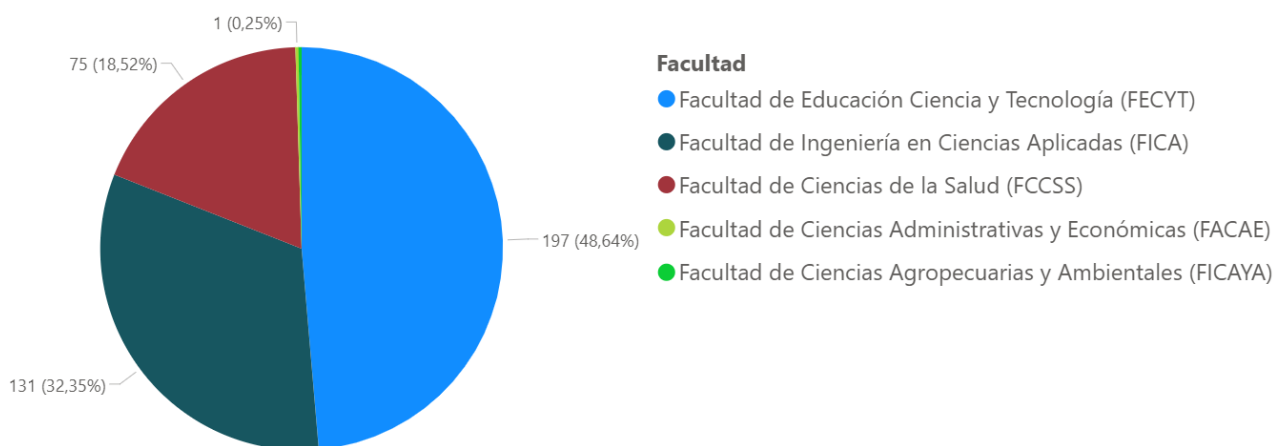


Figura 8 Distribución estudiantes- Encuesta diagnóstica

Fuente: Propia

La encuesta aplicada tanto a docentes como estudiantes incluyó un conjunto de preguntas orientadas a conocer su percepción sobre distintos tipos de notificaciones académicas, la frecuencia de uso de funcionalidades del sistema actual, y sus preferencias respecto al contenido de las notificaciones.

Resultados

En esta sección se presentan los resultados más significativos obtenidos a partir de las encuestas aplicadas a docentes y estudiantes. Estos datos permitieron identificar los tipos de notificaciones académicas más valoradas por los usuarios.

En la **Tabla 5** se muestran las preguntas referentes al tipo de notificaciones de docentes.

Tabla 5 Preguntas docente - Percepción de utilidad en notificaciones

TIPOS DE NOTIFICACIONES		NADA ÚTIL	POCO ÚTIL	MUY ÚTIL
Eventos universitarios generales (Ej.: conferencias, deportivos, etc.)	eventos culturales,	3	26	89
Recordatorio de horario de clase		15	40	63
Recordatorio de registro de asistencia pendiente		4	9	105
Recordatorio de registro de avance pendiente		1	5	112
Recordatorio de tareas pendientes a calificar		4	35	79

En la **Figura 9** se muestra las respuestas tabuladas referente a la **Tabla 5**.

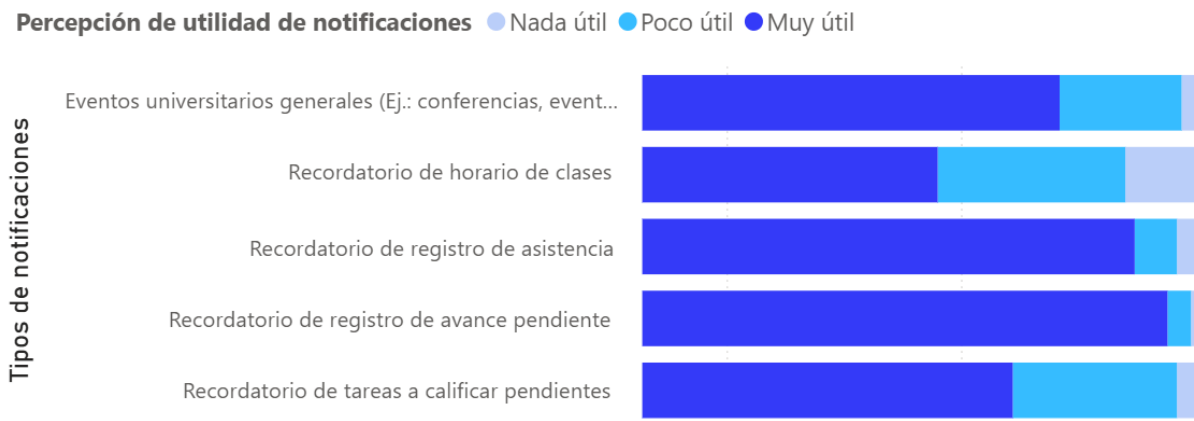


Figura 9 Percepción de utilidad docentes

Fuente: Propia

En la **Tabla 6** se muestran las preguntas referentes al tipo de notificaciones de estudiantes.

Tabla 6 Preguntas estudiante - Percepción de utilidad en notificaciones

TIPOS DE NOTIFICACIONES		MUY ÚTIL	POCO ÚTIL	NADA ÚTIL
Eventos universitarios generales (Ej.: conferencias, eventos deportivos, culturales, etc.)		208	171	26
Horario de clase		170	201	34
Nuevas tareas asignadas		377	25	3
Publicación de notas		325	74	6
Recordatorio de tareas próximas a vencerse		384	20	1

En la **Figura 10** se muestra las respuestas tabuladas referente a la **Tabla 6**.

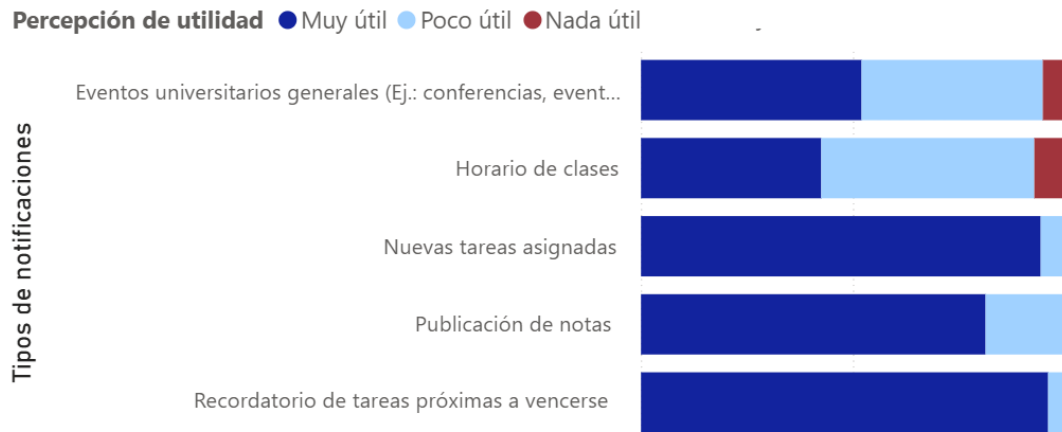


Figura 10 Percepción de utilidad estudiantes

Fuente: Propia

2.1.2.2 Historias de usuario

Tabla 7 Historia de Usuario 1

Historia de Usuario:	Notificación automática por asignación de actividades
ID: HU1	Usuario: estudiante
Descripción:	Como estudiante, quiero recibir notificaciones automáticas cuando se me asignen nuevas actividades (trabajos o evaluaciones), para estar al tanto de mis responsabilidades académicas.
Criterios de aceptación:	<p>CA1: Cuando un docente registra una nueva actividad en el sistema, este dispara un evento que genera una notificación destinada a los estudiantes correspondientes.</p> <p>CA2: La notificación incluye claramente el título de la actividad y la fecha límite de entrega o cumplimiento.</p> <p>CA3: Los estudiantes son suscritos automáticamente al canal de notificaciones de acuerdo con su carrera, paralelo o asignatura correspondiente, sin necesidad de configuraciones manuales.</p> <p>CA4: Las notificaciones se reciben en el cliente móvil en tiempo real o en el momento en que el dispositivo se encuentre activo.</p>

Tabla 8 Historia de Usuario 2

Historia de Usuario:	Notificación automática de registro de notas
ID:HU2	Usuario: estudiante
Descripción:	Como estudiante, quiero recibir una notificación cuando se registren mis notas de evaluaciones o trabajos, para conocer mis resultados de manera oportuna.
Criterios de aceptación:	<p>CA1: Cuando un docente registra una nueva calificación en el sistema, este genera automáticamente una notificación para los estudiantes involucrados.</p> <p>CA2: La notificación incluye el nombre de la asignatura, la actividad y la nota obtenida.</p> <p>CA3: La notificación se envía únicamente al estudiante correspondiente, garantizando la privacidad de la información académica.</p> <p>CA4: El estudiante recibe la notificación en el cliente móvil, en tiempo real o en el momento en que el dispositivo se encuentre activo.</p>

Tabla 9 Historia de Usuario 3

Historia de Usuario:	Notificación automática del horario académico
ID:HU3	Usuario: docente
Descripción:	Como docente, quiero recibir notificaciones automáticas con la información actualizada sobre el horario de clases, incluyendo la instalación física donde se llevará a cabo, para asegurarme de asistir puntualmente a la clase.

Criterios de aceptación:	<p>CA1: El sistema genera una notificación automática entre 5 y 10 minutos antes del inicio de cada clase, según el horario académico configurado para el docente.</p> <p>CA2: Cada notificación contiene el nombre de la asignatura y la ubicación física del aula asignada.</p> <p>CA3: La información concuerda con el horario académico específico del docente, garantizando su precisión y vigencia.</p>
---------------------------------	--

Tabla 10 Historia de Usuario 4

Historia de Usuario:	Notificación Personalizada desde el Sistema Web
ID:HU4	Usuario: administrador web
Descripción:	Como administrador, quiero enviar notificaciones personalizadas a todos los usuarios de la plataforma móvil o a grupos específicos (como docentes o estudiantes de una facultad o carrera), para comunicar eventos importantes o información institucional.
Criterios de aceptación:	<p>CA1: El administrador puede elegir enviar notificaciones a todos los usuarios o segmentarlas por grupos como facultad, carrera o rol (docente, estudiante).</p> <p>CA2: Las notificaciones personalizadas incluyen un título, descripción, categoría y campos opcionales como (url de redirección).</p>

Tabla 11 Historia de Usuario 5

Historia de Usuario:	Suscripción y desuscripción a grupos/topics de usuario
ID:HU5	Usuario: administrador web
Descripción:	Como administrador, quiero suscribir y desuscribir a estudiantes y docentes de los temas de facultades, carreras o roles, para asegurar que reciban solo las notificaciones relevantes.

Criterios de aceptación:	<p>CA1: El administrador puede seleccionar a dispositivos de estudiantes o docentes y suscribirlos o desuscribirlos de temas específicos (por ejemplo, "carreraX", "facultadX", "estudiante", "docente").</p> <p>CA2: El servicio backend realiza las suscripciones o desuscripciones en Firebase para actualizar los temas de FCM de acuerdo con la selección del administrador.</p> <p>CA3: El sistema web confirma al administrador si el proceso de suscripción o desuscripción fue exitoso, mostrando un mensaje de éxito o error.</p>
---------------------------------	--

Tabla 12 Historia de Usuario 6

Historia de Usuario:	Preferencias de notificaciones
ID:HU6	Usuario: estudiante y docente
Descripción:	Como usuario (estudiante o docente), quiero seleccionar las notificaciones que quiero recibir y que no quiero para gestionar solo la información que deseo recibir.
Criterios de aceptación:	<p>CA1: El usuario sea docente o estudiante, puede desmarcar o marcar las notificaciones que desea recibir desde la aplicación móvil.</p> <p>CA2: El usuario solo recibe las notificaciones pertinentes a las preferencias establecidas en la aplicación.</p> <p>CA3: El usuario puede cambiar sus preferencias de notificaciones en todo momento y desde cualquier dispositivo en el que tenga su sesión iniciada.</p>

Tabla 13 Historia de Usuario 7

Historia de Usuario:	Lista de notificaciones
ID:HU7	Usuario: administrador y asistente web

Descripción:	Como usuario (administrador o asistente), quiero ver la lista de notificaciones presentes en la cola de notificaciones para saber el estado de envío en el cual se encuentran las notificaciones agendadas.
Criterios de aceptación:	<p>CA1: Las notificaciones contienen toda la información relevante para el usuario: (id, título, cuerpo, fecha, destino y estado de envío)</p> <p>CA2: El usuario puede aplicar filtros sobre la lista de notificaciones para una mejor búsqueda de información.</p> <p>CA3: La lista de notificaciones se muestra con paginación para optimizar su visualización y desempeño.</p>

Tabla 14 Historio de Usuario 8

Historia de Usuario:	Gestión de notificaciones
ID:HU8	Usuario: administrador web
Descripción:	Como administrador, quiero gestionar el estado de envío de las notificaciones presentes en la lista de notificaciones para poder reagendar una notificación que ha sido cancelada o enviada.
Criterios de aceptación:	<p>CA1: Cada notificación debe incluir un campo adicional que indique su estado de envío, el cual puede ser: <i>Enviado</i>, <i>No enviado</i> o <i>Cancelado</i>.</p> <p>CA2: El estado de cada notificación puede modificarse mediante controles específicos visibles únicamente para usuarios con rol de Administrador web.</p> <p>CA3: Las transiciones permitidas entre estados son las siguientes:</p> <ul style="list-style-type: none"> • De <i>Enviado</i> → a <i>No enviado</i> o <i>Cancelado</i> • De <i>Cancelado</i> → a <i>No enviado</i> o <i>Enviado</i> • De <i>No enviado</i> → a <i>Cancelado</i> o <i>Enviado</i>

CA4: La actualización del estado de una notificación debe reflejarse inmediatamente en la interfaz del sistema de gestión web, incluyendo su nuevo estado en la lista correspondiente.

Tabla 15 Historia de Usuario 9

Historia de Usuario:	Gestión de grupos/topics
ID:HU9	Usuario: administrador web
Descripción:	Como administrador, quiero gestionar los grupos/topics existentes (crear, editar y eliminar) para facilitar la operatividad del sistema y mejorar la administración de los grupos a los que pertenecen los usuarios
Criterios de aceptación:	<p>CA1: La lista de grupos/topics deberá incluir un filtro de búsqueda por nombre, que permita localizar rápidamente grupos específicos.</p> <p>CA2: La visualización de la lista de grupos/topics debe contar con paginación, garantizando un rendimiento adecuado en caso de grandes volúmenes de datos.</p> <p>CA3: El sistema permitirá realizar operaciones de creación, edición y eliminación de grupos/topics directamente desde la interfaz web del administrador.</p>

Tabla 16 Historia de Usuario 10

Historia de Usuario:	Notificaciones de usuario
ID:HU10	Usuario: estudiante y docente
Descripción:	Como usuario (docente o estudiante), quiero visualizar la lista de notificaciones que he recibido para mantenerme informado sobre las novedades universitarias y las actividades correspondientes.

Criterios de aceptación:	CA1: Las notificaciones mostradas son únicamente las que le corresponden al usuario autenticado.
	CA2: El usuario (docente o estudiante), puede marcar cada notificación como leída y puede eliminarlas.
	CA3: La lista de notificaciones del usuario se divide en dos secciones enviadas y no enviadas, según las marque en la interfaz gráfica.

Tabla 17 Historia de Usuario 11

Historia de Usuario:	Notificación Programada desde el Sistema Web
ID:HU11	Usuario: administrador y asistente web
Descripción:	Como usuario (administrador o docente), quiero enviar notificaciones personalizadas a todos los usuarios de la plataforma móvil o a grupos específicos (como docentes o estudiantes de una facultad o carrera) con un campo adicional de fecha programada, para comunicar eventos importantes o información institucional en la fecha establecida.
Criterios de aceptación:	<p>CA1: El administrador puede elegir enviar notificaciones a todos los usuarios o segmentarlas por grupos como facultad, carrera o rol (docente, estudiante).</p> <p>CA2: Las notificaciones personalizadas incluyen un título, descripción y campos opcionales como (url de redirección).</p> <p>CA3: Las notificaciones se envían en la fecha establecida desde el formulario.</p>

Tabla 18 Historia de Usuario 12

Historia de Usuario:	Envío Notificación FCM
ID:HU12	Usuario: sistema

Descripción:	Como sistema, quiero enviar notificaciones mediante FCM para que los usuarios reciban alertas en sus dispositivos móviles.
Criterios de aceptación:	<p>CA1: Las notificaciones se envían desde el backend a uno o varios grupos/topics o a dispositivos individuales según la segmentación configurada.</p> <p>CA2: La notificación incluye un título, cuerpo y datos personalizados definidos en la lógica del evento que la dispara.</p> <p>CA3: El sistema procesa la respuesta de FCM y registra la notificación en la base de datos.</p>

Tabla 19 Historia de Usuario 13

Historia de Usuario:	Suscripciones vía FCM
ID:HU13	Usuario: sistema
Descripción:	Como sistema, quiero suscribir y desuscribir dispositivos a uno o varios topics para facilitar el envío de notificaciones grupales vía FCM.
Criterios de aceptación:	<p>CA1: Se ejecutan llamadas a la API de FCM para realizar la suscripción o desuscripción de uno o más dispositivos a los grupos/topics definidos.</p> <p>CA2: Las operaciones de suscripción o desuscripción están vinculadas al contexto del usuario estudiante o docente.</p> <p>CA3: El sistema registra en base de datos los cambios en las suscripciones</p>

Tabla 20 Historia de Usuario 14

Historia de Usuario:	Recordatorio de registro de asistencia
-----------------------------	---

ID: HU14	Usuario: docente
Descripción:	Como docente, quiero recibir una notificación automática al finalizar una clase si no he registrado la asistencia, para evitar omitir esta tarea.
Criterios de aceptación:	<p>CA1: El backend detecta al final de cada clase si se ha efectuado el registro de asistencia.</p> <p>CA2: Se genera una notificación automática indicando la clase correspondiente.</p> <p>CA3: La notificación permite redirigir al módulo de registro de asistencia.</p>

Tabla 21 Historia de Usuario 15

Historia de Usuario:	Recordatorio de registro de avance
ID: HU15	Usuario: docente
Descripción:	Como docente, quiero recibir una notificación automática al finalizar una clase si no he registrado el avance académico correspondiente, para mantener actualizado el historial de progreso.
Criterios de aceptación:	<p>CA1: El backend detecta al final de cada clase si se ha efectuado el registro de avance.</p> <p>CA2: Se genera una notificación automática indicando la clase correspondiente.</p> <p>CA3: La notificación permite redirigir al módulo de registro de avance.</p>

Tabla 22 Historia de Usuario 16

Historia de Usuario:	Notificación automática de tareas pendientes
-----------------------------	---

ID:HU16	Usuario: estudiante
Descripción:	Como estudiante, quiero recibir una notificación automática cuando una tarea pendiente esté próxima a vencer, para poder completarla y entregarla a tiempo.
Criterios de aceptación:	<p>CA1: Si falta menos de una hora para la fecha límite de una tarea y no se ha entregado aún, el sistema genera una notificación automática dirigida exclusivamente al estudiante correspondiente.</p> <p>CA2: La notificación incluye el nombre de la actividad y la asignatura correspondiente.</p> <p>CA3: La notificación se envía únicamente solo si el estudiante aún no tiene una tarea adjuntada a la actividad.</p>

2.1.3 Product Backlog

El Product Backlog representa la lista de requisitos priorizados necesarios para el desarrollo del proyecto. Este se construyó considerando las necesidades del sistema y las funcionalidades requeridas para la aplicación. Los ítems en el Product Backlog se clasificaron de acuerdo con la complejidad y la prioridad asignada por el equipo de desarrollo.

En la **Tabla 23** se muestran las tareas definidas para el desarrollo del proyecto.

Tabla 23 Product Backlog

Nro.	ID	Nombre	Estimación
1	HU1	Notificación automática por asignación de actividades	5
2	HU2	Notificación automática de registro de notas	5
3	HU3	Notificación automática del horario académico	8
4	HU4	Notificación Personalizada desde el Sistema Web	3
5	HU5	Suscripción y desuscripción a grupos/topics de usuario	3
6	HU6	Preferencias de notificaciones	3

7	HU7	Lista de notificaciones	2
8	HU8	Gestión de notificaciones	3
9	HU9	Gestión de grupos/topics	3
10	HU10	Notificaciones de usuario	5
11	HU11	Notificación Programada desde el Sistema Web	5
12	HU12	Envío Notificación FCM	2
13	HU13	Suscripción vía FCM	2
14	HU14	Recordatorio de registro de asistencia	5
15	HU15	Recordatorio de registro de avance	5
16	HU16	Notificación automática de tareas pendientes	8

2.2 Fase de Juego

2.2.1 Planificación Sprint 0

El Sprint 0 se enfocó en la configuración inicial del proyecto y estableció las bases necesarias para el desarrollo posterior. Este sprint, considerado como de preparación, no incluyó entregables funcionales, pero garantizó que el equipo estuviera preparado para iniciar el desarrollo funcional en los siguientes sprints.

2.2.1.1 Arquitectura general del proyecto

La arquitectura del proyecto se fundamenta en un enfoque basado en microservicios. Para el desarrollo del servicio backend se implementó una API REST utilizando el framework NestJS, seleccionado como resultado del experimento comparativo de tecnologías, y desarrollado en TypeScript. Este servicio está conectado directamente a la base de datos institucional de la UTN, gestionada en Oracle.

En cuanto al desarrollo del frontend móvil se optó por la framework Flutter, basado en el lenguaje de programación Dart. Esta aplicación móvil está registrada en Firebase para el servicio de notificaciones push mediante FCM. Por otro lado, el frontend web, utilizado para la administración del microservicio de notificaciones, fue desarrollado con el framework Next.js, basado en la biblioteca React.

Como se muestra en la **Figura 11**, un API Gateway alojado en un servidor independiente actúa como intermediario entre el cliente (móvil o web) y los distintos microservicios. Su función principal es recibir, autenticar y validar los tokens de identidad del usuario, redireccionando las solicitudes mediante el uso de claves API asignadas a cada microservicio.

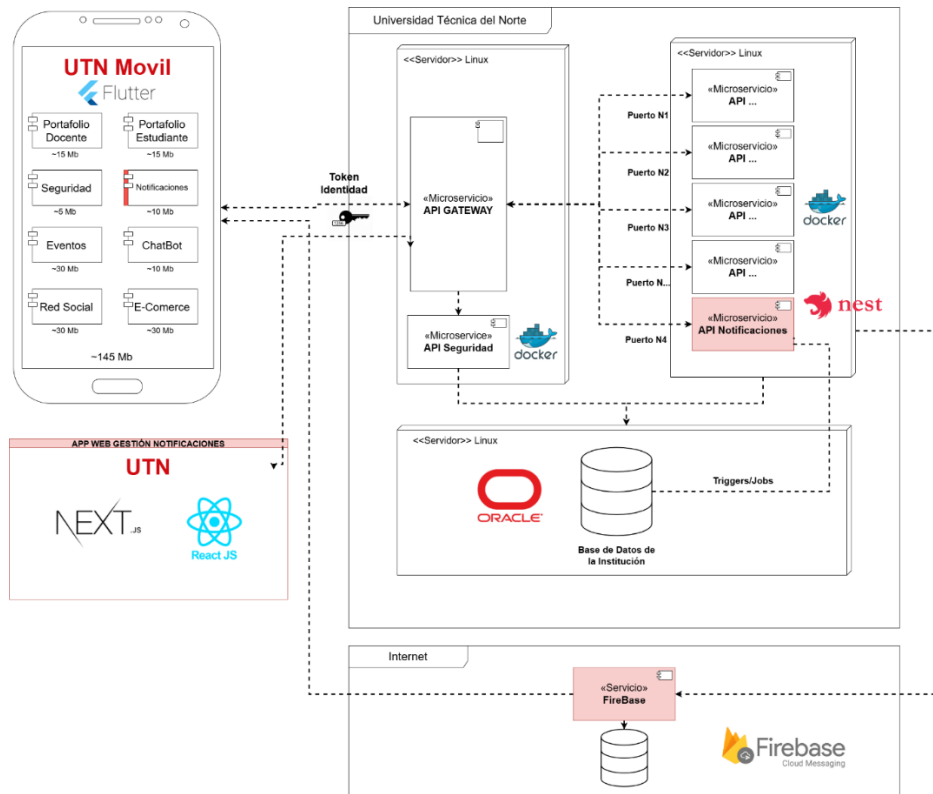


Figura 11 Arquitectura general del proyecto

Fuente: Propia

2.2.1.2 Arquitectura del servicio de notificaciones

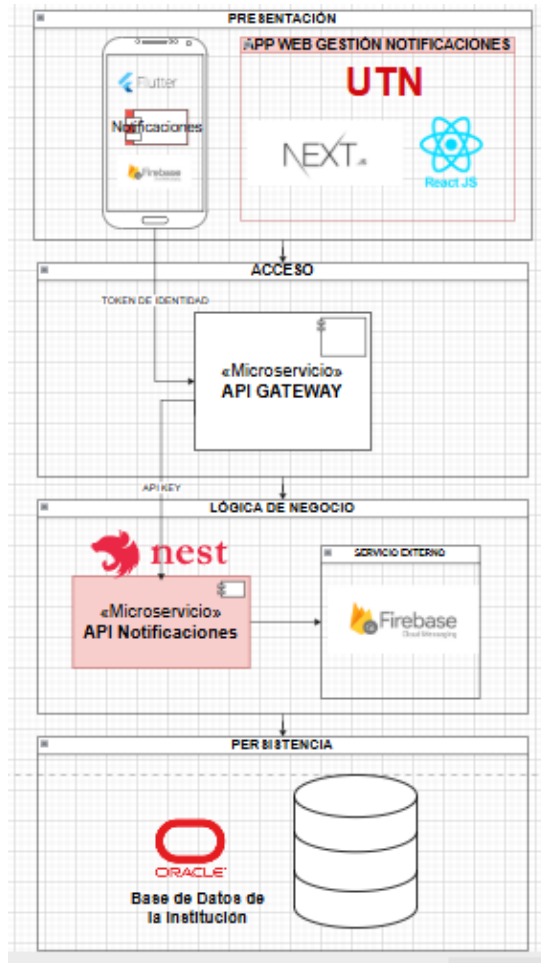


Figura 12 Arquitectura servicio de notificaciones

2.2.1.3 Diagrama de secuencia

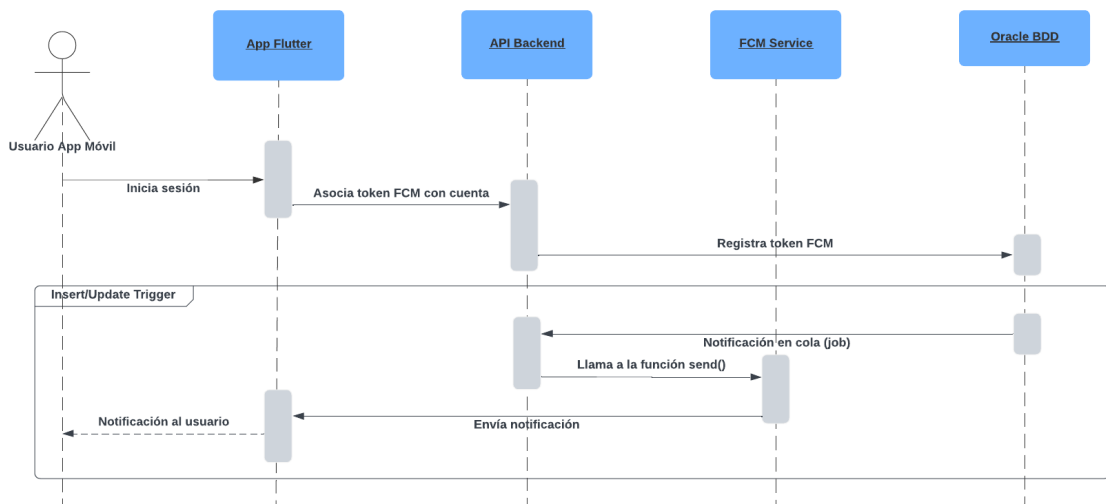
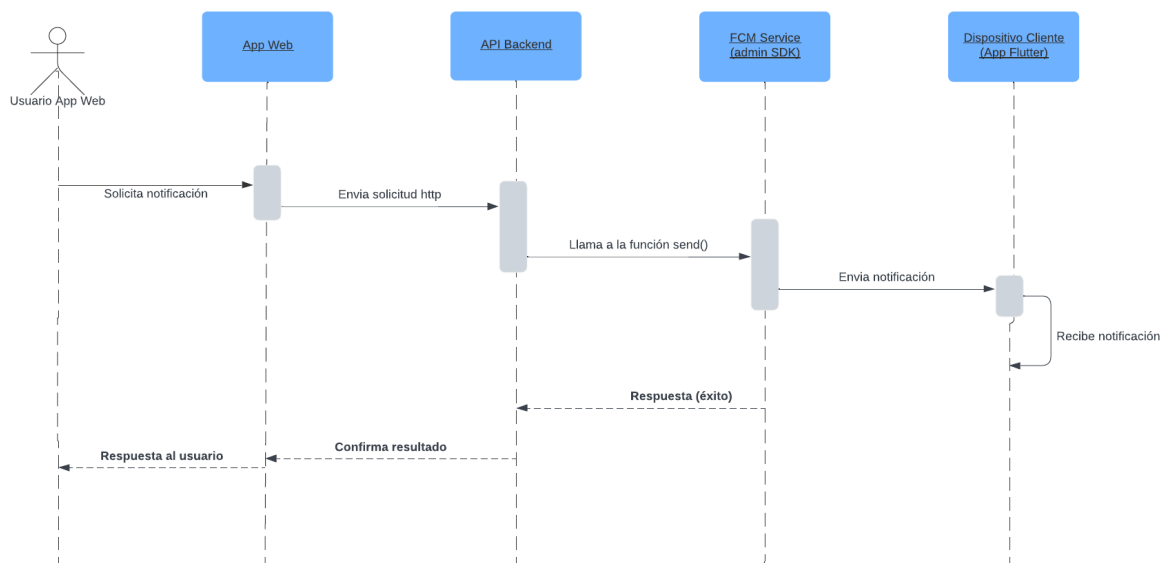


Figura 13 Diagrama de secuencia

Fuente: Propia

2.2.1.4 Configuración del proyecto

Durante la configuración del proyecto se integró el sistema de control de versiones Git, utilizando un repositorio alojado en la plataforma GitHub. Esta herramienta facilitó la gestión del código fuente y la coordinación del desarrollo tanto para el servicio backend como para la interfaz web de administración y la aplicación móvil. Las **Figuras 14, 15 y 16** ilustran la estructura de directorios correspondiente a cada uno de estos proyectos.

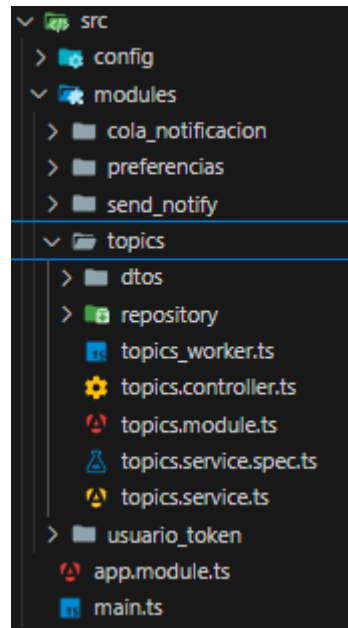


Figura 14 Estructura de directorios backend

La **Figura 14** presenta el contenido del directorio **src**, en el cual se distinguen los directorios principales: **config**, destinado para la gestión de seguridad y acceso a la base de datos y el directorio **modules** el cual presenta las funcionalidades modulares del API REST. Este último se organiza en las siguientes secciones:

- **Dtos:** Contiene los Data Transfer Objects (DTOs), que definen la forma de los datos que se reciben y envían en los endpoints del módulo.
- **Repository:** Contiene la lógica de acceso a la base de datos para el módulo y las consultas SQL.
- **Controller:** Define los endpoints HTTP del módulo, mediante el cual se recibe las peticiones del cliente.
- **Service:** Contiene la lógica de negocio del módulo, gestiona el flujo entre controller, repository y otros servicios.
- **Archivos Spec:** Pruebas unitarias para el service de cada módulo.

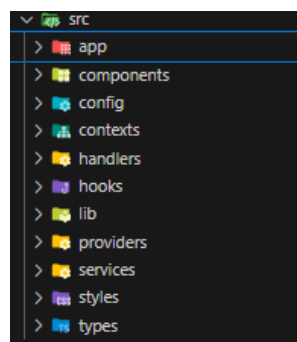


Figura 15 Estructura de directorios aplicación web

La **Figura 15** presenta la estructura del directorio **src** correspondiente al proyecto desarrollado con Next.js, utilizando el enfoque basado en App Router. En el directorio **app** se definen las rutas principales de la aplicación, donde cada subdirectorio representa una ruta específica dentro del sistema.

El directorio **components** contiene elementos de interfaz de usuario reutilizables, como la barra lateral de navegación y componentes específicos para login o protección de rutas. En **config** se centralizan archivos de configuración, como las rutas hacia los endpoints del backend.

De igual manera se usa React Contexts ubicados en el directorio **contexts** para el manejo de estado global, como autenticación. Por otro lado, los directorios **handlers**, **hooks** y **lib**, agrupan funciones personalizadas, lógica reutilizable, librerías y utilidades auxiliares.

La estructura también incorpora el directorio **providers**, donde se definen los proveedores globales de React, y **services**, que concentra la lógica de comunicación con el backend a través de servicios organizados por módulo. Finalmente, los estilos globales del sistema se ubican en el directorio **styles**, y las definiciones de tipos como DTOs y entidades TypeScript se agrupan en **types**, incluyendo estructuras de datos y entidades utilizadas a lo largo de la aplicación.

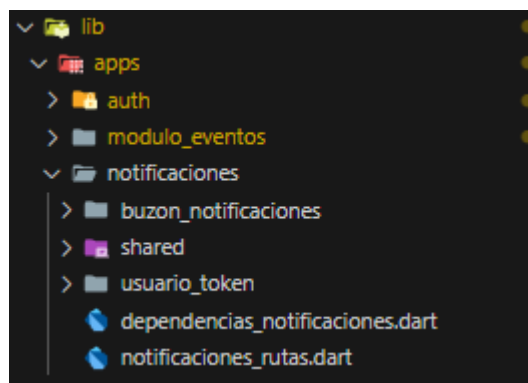


Figura 16 Estructura de directorios aplicación móvil

En la **Figura 16** se presenta la estructura de los directorios referentes a la aplicación móvil desarrollada con Flutter. Dentro de **apps** se encuentra la estructura modular correspondiente a las funcionalidades de la aplicación.

Cada módulo sigue una organización basada en capas, comenzando por el directorio **domain**, que define interfaces, entidades y servicios asociados a la lógica de negocio. A continuación, el directorio **infrastructure** contiene las implementaciones concretas de los

repositorios, así como los modelos de datos utilizados para la comunicación con fuentes externas como las APIs REST. La capa **presentation** agrupa todos los elementos relacionados con la interfaz de usuario, como páginas, widgets y controladores de estado.

Adicionalmente, el módulo cuenta con un directorio **shared** que reúne utilidades y elementos compartidos por las distintas capas, así como una sección dedicada a la configuración de dependencias, permitiendo mantener una separación clara de responsabilidades y una arquitectura flexible y escalable.

2.2.2 Sprint 1

2.2.2.1 Planificación Sprint 1

La planificación del Sprint 1 se centró en definir las tareas prioritarias. Durante esta fase, se seleccionó los ítems del Product Backlog que se consideraron más relevantes y que sirvieron como base para el posterior desarrollo de cada Sprint. El Sprint 1 tenía como objetivo desarrollar la funcionalidad básica de los distintos tipos de envío de notificaciones implementando el servicio de FCM en el área de backend.

En la **Tabla 24**, se detallan las historias de usuario seleccionadas para este Sprint.

Tabla 24 Sprint Backlog 1

ID	NOMBRE	ACTIVIDAD	HORAS
HU12	Envío	Endpoint para el envío a un solo grupo.	1
	Notificación FCM	Endpoint para el envío a varios grupos.	1
		Endpoints para el envío en base a condiciones	2
		Endpoint para el envío a un dispositivo.	1
		Endpoint para el envío a varios dispositivos.	1
HU13	Suscripción vía FCM	Endpoint para la suscripción de un dispositivo a varios grupos.	1
		Endpoint para la suscripción de varios dispositivos a un grupo.	1
		Endpoint para la desuscripción de un dispositivo a varios grupos.	1
		Endpoint para la desuscripción de varios dispositivos a un grupo.	1
HU14	Recordatorio de registro de asistencia	de Desarrollo de funciones para la tarea programada Job dento del backend.	2
		de Desarrollo de funciones para crear las notificaciones en base al horario del docente.	2
HU15	Recordatorio de registro de avance	de Desarrollo de funciones para la tarea programada Job dento del backend.	2
		de Desarrollo de funciones para crear las notificaciones en base al horario del docente.	2

2.2.2.2 Resultado Sprint 1

Los entregables referentes a la historia de usuario HU12, se presentan a continuación en las figuras **Figura 17**, **18** y **19**.

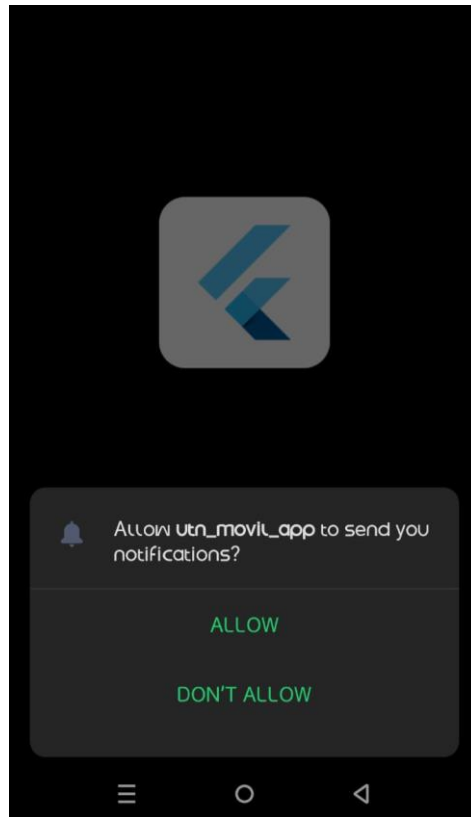


Figura 17 Permisos de notificación en la aplicación

```
import { Injectable, BadRequestException, InternalServerErrorException, Logger } from "@nestjs/common";
import * as admin from "firebase-admin";
import { MultipleDeviceNotificationDto, NotificationDto, TopicNotificationDto } from "../send_notify/dtos/notifica
import { NotificationRepository } from "../repository/send_notify.repository";
import { NotificationQueryDto } from "../dtos/notification-query.dto";
import { UpdateNotificationEnviadoDto } from "../dtos/update-enviado-query.dto";

@Injectable()
export class SendNotifyService {
  private readonly logger = new Logger(SendNotifyService.name);
  constructor(private readonly notificationRepository: NotificationRepository) { }
  async sendNotification(notificationDto: NotificationDto) { ...
  }
  async sendNotificationToMultipleTokens(multipleDeviceNotificationDto: MultipleDeviceNotificationDto) { ...
  }
  async sendTopicNotification(topicNotificationDto: TopicNotificationDto) { ...
  }
  async sendNotificationToMultipleTopics(topicNotificationDto: TopicNotificationDto, topics: string[]) { ...
  }
  async sendNotificationToMultipleTopicsOr(topicNotificationDto: TopicNotificationDto, topics: string[]) { ...
  }
}
```

Figura 18 Servicio para el envío de notificaciones FCM

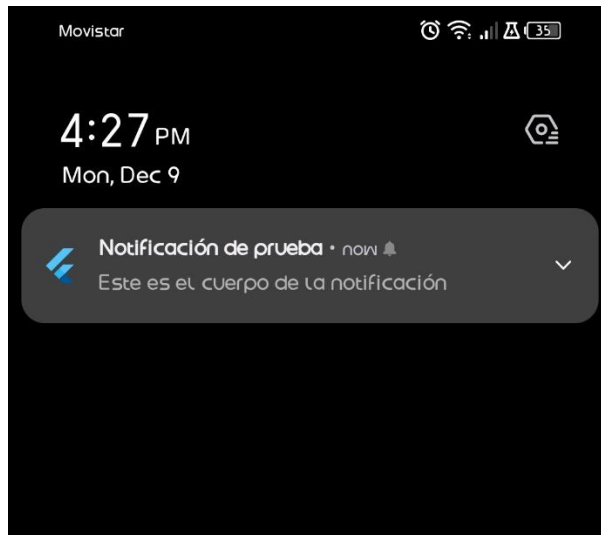


Figura 19 Notificación de prueba vía FCM

Los incrementos referentes a la suscripción y desuscripción de usuarios de HU13 se muestra a continuación en la **Figura 20**.

```
import { Controller, Post, Body, UseGuards, Get, Query, Patch, Delete } from '@nestjsjs/common';
import { TopicSubscriptionService } from '../topics.service';
import { SubscribeToTopicDto, SubscribeToMultipleTopicsDto } from '../dtos/subscribe_topic.dto';
import { AuthGuard } from 'src/config/auth/guard/auth_guard';
import { PaginationDto } from '../dtos/pagination.dto';
import { PermissionGuard } from 'src/config/auth/guard/permission.guard';
import { HasPermission } from 'src/config/auth/decorators/has-permission.decorator';
import { TopicDto } from '../dtos/topic.dto';

@Controller('api/v1/topics-fcm')
export class TopicsFcmController {
  constructor(private readonly topicSubscriptionService: TopicSubscriptionService) {}

  //suscribir tokens a un topic
  @Post("subscribe-multiple-tokens")
  @UseGuards(AuthGuard)
  async subscribeTokensToTopic(@Body() body: SubscribeToTopicDto) {
    return this.topicSubscriptionService.subscribeTokensToTopic(body);
  }

  //desuscribir tokens a un topic
  @Post("unsubscribe-multiple-tokens")
  @UseGuards(AuthGuard)
  async unsubscribeTokensFromTopic(@Body() body: SubscribeToTopicDto) {
    return this.topicSubscriptionService.unsubscribeTokensFromTopic(body);
  }

  @UseGuards(AuthGuard, PermissionGuard)
  @HasPermission("GESTIONAR_SUSCRIPCIONES", "NOTIFICACIONES")
  @Post("subscribe-multiple-topics")
  async subscribeTokenToTopics(@Body() body: SubscribeToMultipleTopicsDto) {
    return this.topicSubscriptionService.subscribeTokenToTopics(body);
  }

  @UseGuards(AuthGuard, PermissionGuard)
  @HasPermission("GESTIONAR_SUSCRIPCIONES", "NOTIFICACIONES")
  @Post("unsubscribe-multiple-topics")
  async unsubscribeTokenFromTopics(@Body() body: SubscribeToMultipleTopicsDto) {
    return this.topicSubscriptionService.unsubscribeTokenFromTopics(body);
  }
}
```

Figura 20 Controller suscripción a grupos/topics FCM

En cuanto a las historias de usuario HU14 y HU15, se desarrollaron validaciones para únicamente entregar las notificaciones de asistencia y avance en docentes, si no se ha realizado un registro, como se presenta en la **Figura 21**.

```

async processColaNotificacion() {
  const pending = await this.colaNotificacionRepository.getNotificacionesPendientes();

  await Promise.allSettled(pending.map(async (notif) => {
    try {
      const clave = notif.CLAVE_NOTIFICACION;

      if (clave !== null && clave !== undefined && clave.trim() !== '') {
        if (clave.startsWith('AVANCE')) {
          const yaRegistradoAvance = await this.colaNotificacionRepository.verificarAvanceRegistradoPorClave(clave);
          if (yaRegistradoAvance) {
            await this.colaNotificacionRepository.cancelarNotificacionPorClave(clave);
            this.logger.warn("Notificación con clave ${clave} ya ha sido registrada, se omite el envío.");
            return;
          }
        } else if (clave.startsWith('ASISTENCIA')) {
          const yaRegistradoAsistencia = await this.colaNotificacionRepository.verificarAsistenciaRegistradaPorClave(clave);
          if (yaRegistradoAsistencia) {
            await this.colaNotificacionRepository.cancelarNotificacionPorClave(clave);
            this.logger.warn("Notificación con clave ${clave} ya ha sido registrada, se omite el envío.");
            return;
          }
        }
      }
    }
  }));
}

```

Figura 21 Servicio para la entrega de notificaciones de avance y asistencia

2.2.2.3 Revisión Sprint 1

Para la revisión de este Sprint se tomó en cuenta que se cumpla con los criterios de aceptación definidos para cada historia de usuario definida, como se puede apreciar en la **Tabla 25**.

Tabla 25 Revisión Sprint 1

ID	CRITERIOS DE ACEPTACIÓN	CUMPLE
HU12	CA1: Las notificaciones se envían desde el backend a uno o varios grupos/topics o a dispositivos individuales según la segmentación configurada.	SÍ
	CA2: La notificación incluye un título, cuerpo y datos personalizados definidos en la lógica del evento que la dispara.	SÍ
	CA3: El sistema procesa la respuesta de FCM y registra la notificación en la base de datos.	SÍ
HU13	CA1: Se ejecutan llamadas a la API de FCM para realizar la suscripción o desuscripción de uno o más dispositivos a los grupos/topics definidos.	SÍ
	CA2: Las operaciones de suscripción o desuscripción están vinculadas al contexto del usuario estudiante o docente.	SÍ
	CA3: El sistema registra en base de datos los cambios en las suscripciones	SÍ

HU14	CA1: El backend detecta al final de cada clase si se ha efectuado el registro de asistencia.	SÍ
	CA2: Se genera una notificación automática indicando la clase correspondiente.	SÍ
	CA3: La notificación permite redirigir al módulo de registro de asistencia.	SÍ
HU15	CA1: El backend detecta al final de cada clase si se ha efectuado el registro de avance.	SÍ
	CA2: Se genera una notificación automática indicando la clase correspondiente.	SÍ
	CA3: La notificación permite redirigir al módulo de registro de avance.	SÍ

2.2.3 Sprint 2

2.2.3.1 Planificación Sprint 2

La planificación del Sprint 2 se centró en las tareas pendientes que surgieron como resultado del Sprint 1. Durante la planificación, se revisó los ítems restantes del Product Backlog y seleccionó aquellos que se podían completar de manera eficiente con respecto al Sprint anterior referente a la funcionalidad backend. Las tareas de este sprint estaban orientadas a completar el funcionamiento del servicio backend del proyecto en conjunto con la integración del nuevo esquema del módulo de notificaciones en la base de datos Oracle.

En la **Tabla 26**, se detallan las historias de usuario seleccionadas para este Sprint.

Tabla 26 Sprint Backlog 2

ID	NOMBRE	ACTIVIDAD	HORAS
HU1	Notificación automática por asignación de actividades	Trigger en la tabla de base de datos al crearse nuevas actividades.	2
		Lógica para filtrar y generar la notificación únicamente para los estudiantes correspondientes.	1
		Lógica para el envío de notificaciones añadidas a la cola de notificaciones.	1
HU2	Notificación automática de registro de notas	Trigger en la tabla de base de datos al actualizar la nota actividades de los estudiantes.	2
		Lógica para filtrar y generar la notificación únicamente para los estudiantes correspondientes.	1
HU3	Notificación automática del horario académico	Desarrollo de funciones para enviar las notificaciones a la cola.	2
		Desarrollo de funciones para crear las notificaciones en base al horario del docente.	2
HU6	Preferencias de notificaciones	Interfaz móvil para selección de tipos de notificación.	1
		Backend para guardar y consultar preferencias	1
		Aplicación de filtros de preferencia en el envío	1
HU10	Notificaciones de usuario	Endpoint para listar notificaciones del usuario	1
		Endpoints para marcar como leídas y eliminar notificaciones	1
		Interfaz móvil para el buzón de notificaciones (leídas y no leídas).	2

2.2.3.2 Resultados Sprint 2

Los entregables referentes a la historia de usuario HU1, se presentan en las **Figuras 22** y **23**.

```
-- Obtener el topic formateado para la materia
SELECT
    'nueva_actividad,' ||
    MOVIL_UTN.MVL_FUN_FORMATEAR_TOPIC_STRING(CA.DESCRIPCION) || '_' ||
    MOVIL_UTN.MVL_FUN_FORMATEAR_TOPIC_STRING(DEP.DESCRIPCION) || '_' ||
    MOVIL_UTN.MVL_FUN_FORMATEAR_TOPIC_STRING(DM.NIVEL_CODIGO) || '_' ||
    MOVIL_UTN.MVL_FUN_FORMATEAR_TOPIC_STRING(MAT.DESCRIPCION) || '_' ||
    MOVIL_UTN.MVL_FUN_FORMATEAR_TOPIC_STRING(DM.PARALELO_CODIGO)
INTO vTopic
FROM
```

Figura 22 Consulta para obtener la materia en trigger

```
if (tipo === 'TOKEN') {
    await this.sendNotification({ token: destino, ...commonPayload });
} else if (tipo === 'MULTITOKEN') {
    const tokens = destino.split(',').map(t => t.trim());
    await this.sendNotificationToMultipleTokens({ tokens, ...commonPayload });
} else if (tipo === 'TOPIC') {
    await this.sendTopicNotification({ topic: destino, ...commonPayload });
} else if (tipo === 'MULTITOPIC') {
    await this.sendNotificationToMultipleTopics({ topics: destino.split(',').map(t => t.trim()), ...commonPayload });
}
else if (tipo === 'TOPIC_CONDICION_ALL') {
    const topics = destino.split(',').map(t => t.trim());
    await this.sendTopicNotificationWithAllConditions({ topics, ...commonPayload });
}
else {
    return { success: false, message: `Tipo de notificación no soportado: ${tipo}` };
}

// Marcar como enviada
await this.colaNotificacionRepository.updateNotificacionEnviada(notId);

// Guardar en MVL_TAB_NOTIFICACIONES_USUARIO después de enviar y marcar como enviado
if (tipo === 'TOKEN' || tipo === 'MULTITOKEN') {
    const tokens = destino.split(',').map((t) => t.trim());
    await Promise.allSettled(
        tokens.map((token) =>
            this.notificacionRepository.saveUserNotification(notId, token)
        )
    );
} else if (tipo === 'TOPIC' || tipo === 'MULTITOPIC') {
    const topics = destino.split(',').map((t) => t.trim());
    await Promise.allSettled(
        topics.map((topic) =>
```

Figura 23 Lógica para el envío de notificaciones en cola

De igual manera, en la **Figura 24**, se puede apreciar que para el trigger relacionado a la HU2, se definió el cuerpo del mensaje.

```

-- Construir el topic
vTopic := 'actividad_calificada,E' || :NEW.CEDULA_INTEGRANTE;

-- Construir título y cuerpo del mensaje
vTitulo := 'Nueva nota asignada en tu materia ' || vNombreMateria;
vCuerpo := 'Tu actividad "' || vNombreActividad || '" ha sido calificada con ' || :NEW.CALIFICACION;

```

Figura 24 Variables dentro del trigger de calificaciones

Para el desarrollo de notificaciones correspondientes al horario de los docentes se definieron las siguientes funciones mostradas en la **Figura 25**.

```

Future<void> _programarNotificacionesDocente() async {
  for (final clase in horarioDocenteProvider.horario) {
    if (clase.inicioHora.isEmpty) continue;

    final fecha = _calcularFecha(
      clase.diaSemanaCodigo, clase.inicioHora, clase.fechaInicio);

    if (fecha != null) {
      final horaInicio = tz.TZDateTime.from(fecha, tz.local);
      final bool esClasePersonal = clase.nivel == '00';

      if (_recordatoriosHorario) {
        if (!esClasePersonal) {
          final fechaNotif = horaInicio.subtract(const Duration(minutes: 10));
          await _enviarRecordatorioDocente(
            "HORARIO",
            "Próxima clase",
            '${clase.materia} está por comenzar en ${clase.estructuraFisica}.',
            "/docente/horarios-docente",
            cuentaDocente,
            fechaNotif,
            clase.materia);
        }
      }
    }
  }
}

```

Figura 25 Función para notificaciones de horario

En cuanto al desarrollo de la interfaz móvil, se elaboró la siguiente vista referente al buzón de notificaciones de docentes y estudiantes, como se aprecia en la **Figura 26** y **27**.

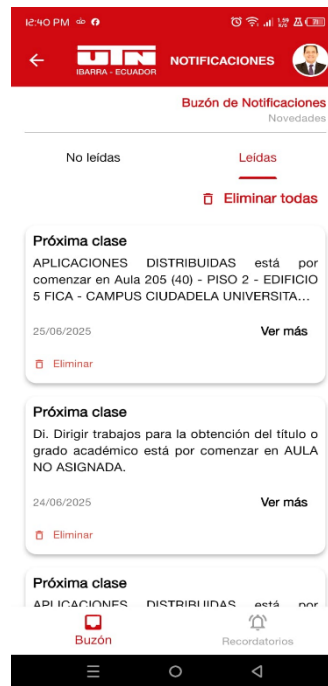


Figura 27 Buzón de notificaciones (no leídas) Figura 26 Buzón de notificaciones (leídas)

De acuerdo con la **Figura 28**, la sección de preferencias de notificaciones de docentes se encuentran las siguientes opciones.

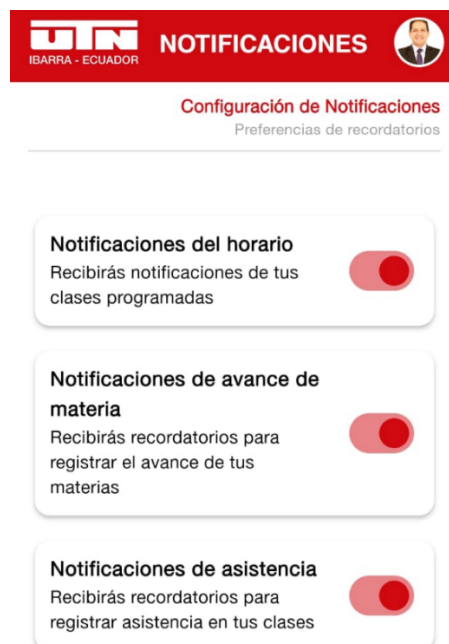


Figura 28 Preferencias de docentes

Para la sección de preferencias de notificaciones, en el caso de que el usuario que haya iniciado sesión tenga el rol de estudiante, se muestran las siguientes funciones en las **Figuras 29 y 30**.

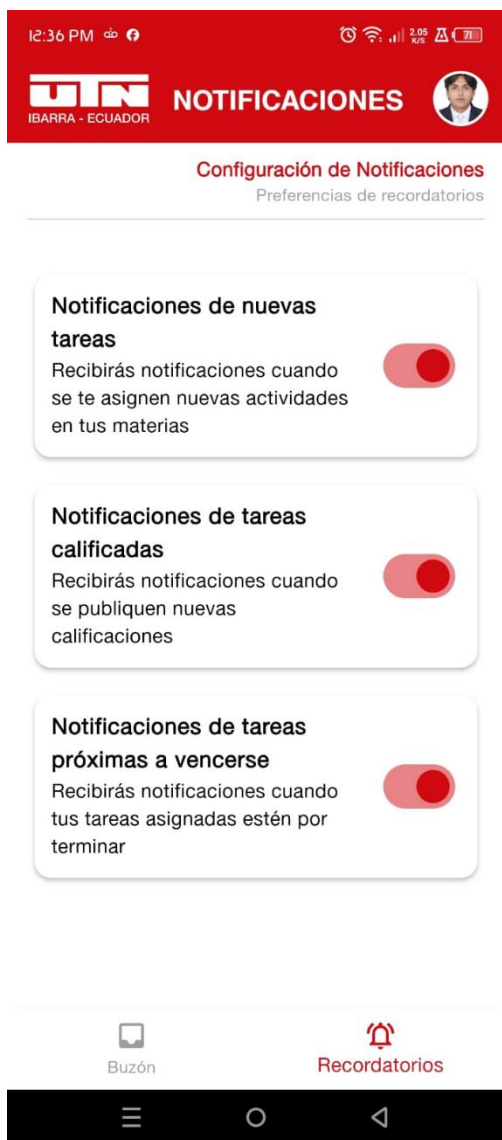


Figura 30 Preferencias de estudiantes

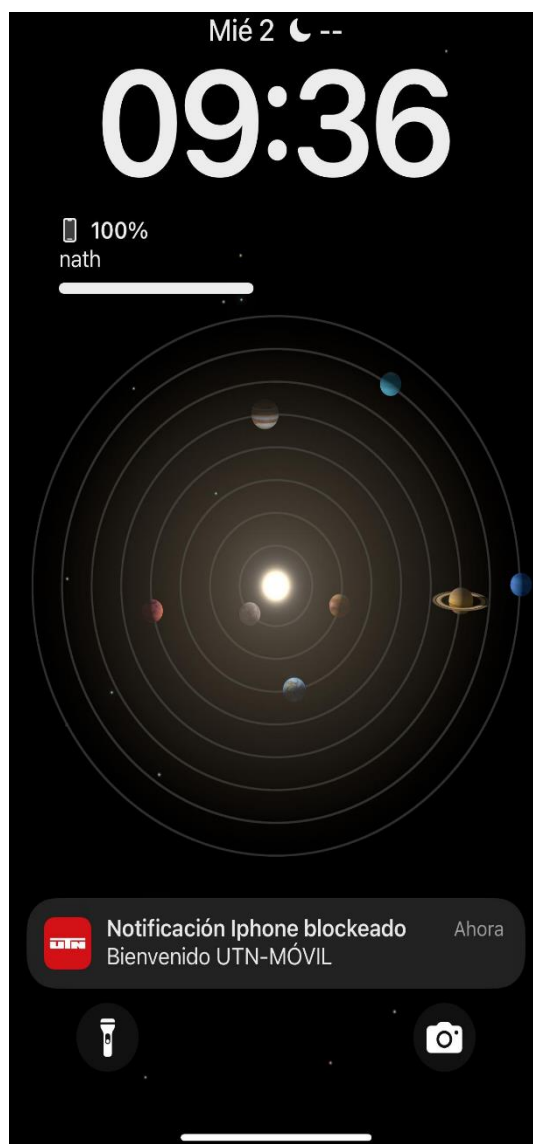


Figura 29 Notificación push en iOS

2.2.3.3 Revisión Sprint 2

Durante la revisión del Sprint se verificó que cada historia de usuario cumpliera con los criterios de aceptación previamente establecidos, como se puede apreciar en la **Tabla 27**.

Tabla 27 Revisión Sprint 2

ID	CRITERIOS DE ACEPTACIÓN	CUMPLE
HU1	CA1: Cuando un docente registra una nueva actividad en el sistema, este dispara un evento que genera una notificación destinada a los estudiantes correspondientes.	Sí
	CA2: La notificación incluye claramente el título de la actividad y la fecha límite de entrega o cumplimiento.	Sí
	CA3: Los estudiantes son suscritos automáticamente al canal de notificaciones de acuerdo con su carrera, paralelo o asignatura correspondiente, sin necesidad de configuraciones manuales.	Sí
	CA4: Las notificaciones se reciben en el cliente móvil en tiempo real o en el momento en que el dispositivo se encuentre activo.	Sí
HU2	CA1: Cuando un docente registra una nueva calificación en el sistema, este genera automáticamente una notificación para los estudiantes involucrados.	Sí
	CA2: La notificación incluye el nombre de la asignatura, la actividad y la nota obtenida.	Sí
	CA3: La notificación se envía únicamente al estudiante correspondiente, garantizando la privacidad de la información académica.	Sí
	CA4: El estudiante recibe la notificación en el cliente móvil, en tiempo real o en el momento en que el dispositivo se encuentre activo.	Sí
HU3	CA1: El sistema genera una notificación automática entre 5 y 10 minutos antes del inicio de cada clase, según el horario académico configurado para el docente.	Sí
	CA2: Cada notificación contiene el nombre de la asignatura y la ubicación física del aula asignada.	Sí
	CA3: La información concuerda con el horario académico específico del docente, garantizando su precisión y vigencia.	Sí
HU6	CA1: El usuario sea docente o estudiante, puede desmarcar o marcar las notificaciones que desea recibir desde la aplicación móvil.	Sí

	CA2: El usuario solo recibe las notificaciones pertinentes a las preferencias establecidas en la aplicación.	Sí
	CA3: El usuario puede cambiar sus preferencias de notificaciones en todo momento y desde cualquier dispositivo en el que tenga su sesión iniciada.	Sí
HU10	CA1: Las notificaciones mostradas son únicamente las que le corresponden al usuario autenticado.	Sí
	CA2: El usuario (docente o estudiante), puede marcar cada notificación como leída y puede eliminarlas.	Sí
	CA3: La lista de notificaciones del usuario se divide en dos secciones enviadas y no enviadas, según las marque en la interfaz gráfica.	Sí

2.2.4 Sprint 3

2.2.4.1 Planificación Sprint 3

La planificación del Sprint 3 se enfocó en el desarrollo del cliente web destinado a la gestión de notificaciones. Durante esta etapa, se seleccionaron del Product Backlog los ítems relacionados con la interfaz de usuario y la integración del frontend web con los servicios previamente implementados, como se muestra en la **Tabla 28**.

Tabla 28 Sprint Backlog 3

ID	NOMBRE	ACTIVIDAD	HORAS
HU4	Notificación Personalizada desde el Sistema	Desarrollo del servicio para invocar al backend y realizar el envío de notificaciones	1
	Web	Formulario web para el envío de notificaciones	2
HU5	Suscripción y desuscripción de grupos/topics de usuario	Desarrollo del servicio para invocar al backend y gestionar las suscripciones de dispositivos a grupos/topics.	1
		Vista con la lista de dispositivos	1
		Vista con la lista de cuentas asociadas a dispositivos	1
		Formulario web para la gestión de suscripción y desuscripción de dispositivos a grupos/topics.	1
HU7	Lista de notificaciones	Desarrollo del servicio para invocar al backend y obtener la información de las notificaciones	1
HU8		Vista con la lista de notificaciones	1
	Gestión de notificaciones	Desarrollo del servicio para invocar al backend y editar el estado de las notificaciones	1
HU9		Modal para editar el estado (enviado, no enviado y cancelado) en la lista de notificaciones.	1
	Gestión de grupos/topics	Desarrollo del servicio para invocar al backend y crear, editar o eliminar grupos/topics.	1
		Vista con la lista de grupos/topics	1
		Modales para editar, crear y eliminar grupos/topics.	1

2.2.4.2 Resultados Sprint 3

Para el envío de notificaciones personalizadas siguiendo la HU4, se implementó el siguiente formulario web con los campos requeridos para hacer la petición HTTP al servicio backend, como se presenta en la **Figura 31**.

UTN-MÓVIL NOTIFICACIONES

Dashboard

Notificaciones

Enviar notificación

Programar notificación

Lista de notificaciones

Grupos

b bilfloresa@utn.edu.ec

Cerrar sesión

Tipo de Notificación

Un solo grupo Múltiples grupos

Selección de Grupo(s)

Seleccione un grupo

software x

Contenido de la Notificación

Categoría

Académica

Título

Integración Carrera de Software

31 / 200 caracteres

Mensaje

Figura 31 Formulario web para el envío de notificaciones

Con respecto a la HU5, se definieron las siguientes vistas con el objetivo apoyar en la gestión de suscripciones de los dispositivos asociados a la aplicación móvil, como se aprecia en las **Figuras 32, 33 y 34**.

UTN-MÓVIL NOTIFICACIONES

Dashboard

Notificaciones

Grupos

Lista de grupos

Dispositivos asociados a usuario

Dispositivos para notificaciones

b bilfloresa@utn.edu.ec

Cerrar sesión

Dispositivos asociados a usuario para notificaciones

Buscar cuenta

Q Buscar por cuenta ...

ID	CUENTA	TOKEN	DISPOSITIVO	FECHA ASOCIACIÓN	ESTADO	ACCIONES
244	E1050258431	dPzx_eC4 SLaylsbtd Vrz... Ver más	Infinix HOT 30 - Infinix X6831	29/6/2025, 12:41:14 p.m.	Activo	Ver Suscripciones Inactivar
246	D1002322384	dPzx_eC4 SLaylsbtd Vrz... Ver más	Infinix HOT 30 - Infinix X6831	29/6/2025, 12:40:01 p.m.	Inactivo	Ver Suscripciones
1282	E1004733695	dvVFlwDt QMajBfX ko-nR... Ver más	Sebas Q - SM -A525M	29/6/2025, 8:57:17 a.m.	Activo	Ver Suscripciones Inactivar
1182	E1004733695	d_vfcm4 TxOvsgU eZDF...	Sebas Q - SM -A525M	29/6/2025, 8:40:38 a.m.	Activo	Ver Suscripciones Inactivar

Figura 32 Vista web de dispositivos asociados a usuarios

UTN-MÓVIL NOTIFICACIONES

Dashboard

Notificaciones

Grupos

Lista de grupos

Dispositivos asociados a usuario

Dispositivos para notificaciones

bilfloresa@utn.edu.ec

Cerrar sesión

Dispositivos para notificaciones

Buscar dispositivo

Q Buscar por token o dispositivo

ID	TOKEN	DISPOSITIVO	FECHA ACTUALIZACIÓN	ACCIONES
1429	cFmGeQuT0uV_I5yPzH h... Ver más		29/6/2025, 6:51:57 p.m.	Ver Suscripciones
3165	cW_MD35-T7CAMqmYvV HK... Ver más	Redmi Note 10S - M2101K7BL	29/6/2025, 6:34:35 p.m.	Ver Suscripciones
2906	fgWeBO6ZQvuLWF9YyMp V... Ver más	A55 de Jordi - SM-A556E	29/6/2025, 3:27:19 p.m.	Ver Suscripciones
494	dPzx_eC4SLaysbtdVrz... Ver más	Infinix HOT 30 - Infinix X6831	29/6/2025, 2:33:08 p.m.	Ver Suscripciones
3166	dn6weeQAR7CIJOy_YQ0 o... Ver más	Redmi Note 8 Pro - Redmi Note 8 Pro	29/6/2025, 2:22:42 p.m.	Ver Suscripciones

Figura 33 Vista web de dispositivos de la aplicación UTN-Móvil

UTN-MÓVIL NOTIFICACIONES

Dashboard

Notificaciones

Grupos

Lista de grupos

Dispositivos asociados a usuario

Dispositivos para notificaciones

bilfloresa@utn.edu.ec

Suscripciones del dispositivo

Dispositivo: Infinix HOT 30 - Infinix X6831
Token: dPzx_eC4SLaysbtdVrzpFAPA91bEgUldJCaODCPoOfOmou3P8Sv0WIZC7bQtXEKs7e8A4LI2XwaTcU537bRD3fh3o8SqwQ10zCg5HFGRFfnjNqjS0K7AsMLNkCCZlFGCrhKPI6SFJts

Grupos disponibles

software

Seleccionados para suscribir:

- software

<< < Página 1 de 10 > >> (98 resultados)

Suscribir seleccionados

Topic
<input checked="" type="checkbox"/> software
<input type="checkbox"/> mar2025ago2025_software_03_ecuacionesdiferenciales_a
<input type="checkbox"/> mar2025ago2025_software_04_probabilidadyestadistica_a
<input type="checkbox"/> mar2025ago2025_software_02_procesosdesoftware_a
<input type="checkbox"/> mar2025ago2025_software_08_gestiondetecnologiasdelainformac
<input type="checkbox"/> mar2025ago2025_tecnologiasdelainformacionenlinea_06_ingenier
<input type="checkbox"/> mar2025ago2025_software_03_herramientasdeprogramacion_a

Grupos suscritos

Seleccionados para desuscribir:

- estudiante

Desuscribir seleccionados

Grupo
<input checked="" type="checkbox"/> estudiante
<input type="checkbox"/> E1050258431
<input type="checkbox"/> nueva_actividad
<input type="checkbox"/> tarea_por_vencer
<input type="checkbox"/> actividad_calificada
<input type="checkbox"/> mar2025ago2025_software_08_titulacion2_a

Figura 34 Vista web para gestión de suscripciones

Asimismo, para la gestión y visualización de las notificaciones existentes, se elaboraron las siguientes vistas dentro de la aplicación web, de acuerdo con las **Figuras 35 y 36**.

UTN-MÓVIL NOTIFICACIONES

Historial de Notificaciones

Buscar notificaciones: Categoría: Fecha inicio: Fecha fin:

ID	TÍTULO / CONTENIDO	CATEGORÍA	DESTINO	FECHA CREACIÓN	ESTADO	ACCIONES
1138	Registro de avance Tienes un avance pendiente de la clase INGENIERÍA DE SOFTWARE... Ver más	ACADEMICA	D1002322384	26/06/2025, 21:00	Enviado	<input type="button" value="Cambiar estado"/>
1139	Asistencia de clase Registra la asistencia pendiente de la clase INGENIERÍA DE S... Ver más	ACADEMICA	D1002322384	26/06/2025, 21:00	Enviado	<input type="button" value="Cambiar estado"/>
1137	Próxima clase INGENIERÍA DE SOFTWARE II está por comenzar en AULA NO ASIGN... Ver más	ACADEMICA	D1002322384	26/06/2025, 17:50	Enviado	<input type="button" value="Cambiar estado"/>

Figura 35 Vista con la lista de notificaciones

26/06/2025, 21:00

26/06/2025, 21:00

Enviado

No enviado

Cancelado

Figura 36 Modal para la gestión del estado de las notificaciones

Finalmente, para la visualización y gestión de los grupos/topics tanto a través de FCM como de la base de datos, se desarrolló la siguiente vista observable en la **Figura 37**.

UTN-MÓVIL NOTIFICACIONES

Gestión de Grupos

Buscar grupo:

ID	NOMBRE	FECHA CREACIÓN	ACCIONES
66741	software	25/06/2025, 08:44	<input type="button" value="✎"/> <input type="button" value="🗑️"/>
44521	sep2023feb2024vincul... Ver más	18/06/2025, 21:35	<input type="button" value="✎"/> <input type="button" value="🗑️"/>
44529	abr2024ago2024vincul... Ver más	18/06/2025, 21:35	<input type="button" value="✎"/> <input type="button" value="🗑️"/>
44528	abr2024jul2025pos_ma... Ver más	18/06/2025, 21:35	<input type="button" value="✎"/> <input type="button" value="🗑️"/>
44527	abr2024ago2024vincul... Ver más	18/06/2025, 21:35	<input type="button" value="✎"/> <input type="button" value="🗑️"/>

Figura 37 Vista web para la gestión de grupos/topics

2.2.4.3 Revisión Sprint 3

Durante la etapa de revisión del Sprint, se llevó a cabo un proceso de verificación orientado a constatar que cada una de las historias de usuario incluidas en el Sprint cumpliera satisfactoriamente con los criterios de aceptación que habían sido definidos con anterioridad.

A continuación, en la **Tabla 29**, se detalla la revisión del Sprint 3.

Tabla 29 Revisión Sprint 3

ID	CRITERIOS DE ACEPTACIÓN	CUMPLE
HU4	CA1: El administrador puede elegir enviar notificaciones a todos los usuarios o segmentarlas por grupos como facultad, carrera o rol (docente, estudiante).	SÍ
	CA2: Las notificaciones personalizadas incluyen un título, descripción, categoría y campos opcionales como (url de redirección).	SÍ
HU5	CA1: El administrador puede seleccionar a dispositivos de estudiantes o docentes y suscribirlos o desuscribirlos de temas específicos (por ejemplo, "carreraX", "facultadX", "estudiante", "docente").	SÍ
	CA2: El servicio backend realiza las suscripciones o desuscripciones en Firebase para actualizar los temas de FCM de acuerdo con la selección del administrador.	SÍ
	CA3: El sistema web confirma al administrador si el proceso de suscripción o desuscripción fue exitoso, mostrando un mensaje de éxito o error.	SÍ
HU7	CA1: Las notificaciones contienen toda la información relevante para el usuario: (id, título, cuerpo, fecha, destino y estado de envío)	SÍ
	CA2: El usuario puede aplicar filtros sobre la lista de notificaciones para una mejor búsqueda de información.	SÍ
	CA3: La lista de notificaciones se muestra con paginación para optimizar su visualización y desempeño.	SÍ
HU8	CA1: Cada notificación debe incluir un campo adicional que indique su estado de envío, el cual puede ser: <i>Enviado</i> , <i>No enviado</i> o <i>Cancelado</i> .	SÍ

HU9	CA2: El estado de cada notificación puede modificarse mediante controles específicos visibles únicamente para usuarios con rol de Administrador web.	SÍ
	CA3: Las transiciones permitidas entre estados son las siguientes: <ul style="list-style-type: none"> • De <i>Enviado</i> → a <i>No enviado</i> o <i>Cancelado</i> • De <i>Cancelado</i> → a <i>No enviado</i> o <i>Enviado</i> • De <i>No enviado</i> → a <i>Cancelado</i> o <i>Enviado</i> 	SÍ
	CA4: La actualización del estado de una notificación debe reflejarse inmediatamente en la interfaz del sistema de gestión web, incluyendo su nuevo estado en la lista correspondiente.	SÍ
	CA1: La lista de grupos/topics deberá incluir un filtro de búsqueda por nombre, que permita localizar rápidamente grupos específicos.	SÍ
	CA2: La visualización de la lista de grupos/topics debe contar con paginación, garantizando un rendimiento adecuado en caso de grandes volúmenes de datos.	SÍ
	CA3: El sistema permitirá realizar operaciones de creación, edición y eliminación de grupos/topics directamente desde la interfaz web del administrador.	SÍ

2.2.5 Sprint 4

2.2.5.1 Planificación Sprint 4

La planificación del Sprint 4 se centró en la culminación de las funcionalidades pendientes y en la ejecución de pruebas funcionales sobre los componentes previamente desarrollados. Durante esta etapa, se priorizaron ítems del Product Backlog que complementaban las características ya implementadas, asegurando la completitud de los requisitos funcionales. Además, se llevaron a cabo tareas orientadas a la validación del comportamiento del sistema que permitieran verificar el cumplimiento de los criterios de aceptación definidos, contribuyendo así en el desarrollo del producto antes de su entrega final.

A continuación, se presenta el Sprint Backlog 4 correspondiente a la **Tabla 30**.

Tabla 30 Sprint Backlog 4

ID	NOMBRE	ACTIVIDAD	HORAS
HU11	Notificación Programada desde el Sistema Web	Desarrollo del servicio para invocar al backend y realizar el envío de notificaciones programadas.	2
		Formulario web para el envío de notificaciones programadas.	2
HU16	Notificación automática de tareas pendientes	Consulta SQL para obtener todas las tareas de los usuarios, que se encuentren pendientes (Sin entrega de adjunto)	1
		Desarrollo de funciones para la tarea programada Job dentro del backend.	3

2.2.5.2 Resultados Sprint 4

Como parte del cumplimiento de la historia de usuario HU11, se desarrolló e integró un formulario web específico que contiene los campos necesarios para emitir una solicitud HTTP hacia el servicio backend, permitiendo así la ejecución de notificaciones programadas, como se muestra en la **Figura 38**:

The screenshot shows a mobile application interface for scheduling notifications. On the left is a dark sidebar with the title 'UTN-MÓVIL NOTIFICACIONES' and menu items: Dashboard, Notificaciones (with a dropdown arrow), Enviar notificación, Programar notificación (highlighted in red), Lista de notificaciones, Grupos (with a right arrow), and Cerrar sesión. The main content area contains the following fields:

- Olimpiadas UTN**: Text input field with a character count of 15 / 200.
- Mensaje**: Text area with the placeholder text 'Se realiza la invitación a toda la comunidad universitaria para formar parte' and a character count of 76 / 500.
- URL (opcional)**: Text input field with the value 'https://ejemplo.com' and a character count of 0 / 200.
- Ruta para app móvil (opcional)**: Text input field with the value '/ruta/app' and a character count of 0 / 100.
- Programar Envío**: Section containing:
 - Fecha**: Date picker set to 29/06/2025.
 - Hora**: Time picker set to 22:52.
- Confirmation message**: A blue banner at the bottom states: 'La notificación se enviará el 29/06/2025 a las 22:52. Asegúrese de que la fecha y hora sean correctas antes de programar la notificación.'

Figura 38 Formulario web para programar notificaciones

Asimismo, en el marco del cumplimiento de la historia de usuario HU16, se implementaron funciones programadas específicas cuyo propósito fue la construcción dinámica del contenido de las notificaciones asociadas a tareas pendientes dirigidas a los estudiantes, como se observa en las **Figuras 39 y 40**.

```
import { Injectable } from '@nestjs/common';
import { Cron, CronExpression } from '@nestjs/schedule';
import { SendNotifyService } from './send_notify.service';

@Injectable()
export class TareasPendientesJob {
  constructor(private readonly notifyService: SendNotifyService) { }

  @Cron(CronExpression.EVERY_HOUR)
  async checkTareasPendientes() {

    await this.notifyService.processTareasPendientes();
  }
}
```

Figura 39 Job programado para la notificación de tareas pendientes

```
async processTareasPendientes() {
  const tareas = await this.notificationRepository.getTareasPorVencerPendientes();

  for (const tarea of tareas) {
    const clave = tarea.ID_ACTIVIDAD_ALUMNO;

    const alreadyExists = await this.notificationRepository.existsByClave(clave);
    if (alreadyExists) continue;

    const topics = ['tarea_por_vencer', `E${tarea.CEDULA_INTEGRANTE}`];
    const title = `Tienes una tarea pendiente`;
    const body = `La actividad "${tarea.NOMBRE_ACTIVIDAD}" de tu materia "${tarea.NOMBRE_MATERIA}" está por vencer.`;

    const notificationDto: TopicNotificationDto = {
      title,
      body
    };

    await this.sendNotificationToMultipleTopics(
      notificationDto,
      topics,
      clave
    );
  }
}
```

Figura 40 Función para el envío de notificaciones de tareas pendientes

III. VALIDACIÓN

3.1 Modelo de calidad en uso

Para evaluar la calidad en uso del servicio de notificaciones y eventos implementado en la aplicación móvil de la Universidad Técnica del Norte, se adoptó como modelo de referencia la norma ISO/IEC 25022. Esta norma forma parte del conjunto de estándares de calidad ISO/IEC 25000 (SQuaRE) y se enfoca en la medición de la calidad en uso, es decir, la experiencia del usuario al interactuar con el sistema [31].

La evaluación consideró las siguientes **características principales** de calidad en uso:

- **Eficacia:** Evalúa si los usuarios logran alcanzar sus objetivos con precisión y completitud.
- **Eficiencia:** Mide los recursos utilizados (tiempo, esfuerzo) en relación con la precisión y completitud con que se logran los objetivos.
- **Satisfacción:** Refleja el grado de comodidad y aceptación del sistema por parte del usuario.

Como se muestra en la **Tabla 31**, cada medida fue detallada dentro del instrumento de evaluación, y ponderada con un peso específico para su posterior análisis cuantitativo.

Tabla 31 Modelo de Calidad en Uso

<i>Característica</i>	<i>Medida</i>	<i>Peso de característica</i>	<i>Peso de medida</i>
<i>Eficacia</i>	Tareas completadas		16%
	Objetivos logrados	42%	16%
	Tareas sin errores		10%
<i>Eficiencia</i>	Tiempo de tareas		16%
	Eficiencia del tiempo	32%	16%
<i>Satisfacción</i>	Utilidad percibida		10%
	Confianza	26%	8%
	Comodidad		8%

3.2 Muestra poblacional

El público objetivo para la aplicación de la encuesta y la realización del taller práctico fue seleccionado mediante un muestreo no probabilístico. La muestra estuvo conformada por 47 estudiantes de los niveles superiores de la carrera de Software de la Facultad de Ingeniería en Ciencias Aplicadas (FICA), como se detalla en el ANEXO A. De este grupo, se identificaron y seleccionaron 5 usuarios expertos, quienes ya cuentan con experiencia previa en el uso de la aplicación UTN-MÓVIL y con el sistema web SIIU de la universidad.

3.3 Instrumentos de evaluación

Para la recolección de datos y la validación de la calidad en uso del sistema, se aplicó una encuesta y taller práctico estructurados de la siguiente manera:

- **Sistema de Escala de Usabilidad (SUS):** Una escala estandarizada de 10 ítems diseñada para medir la usabilidad general del sistema desde la perspectiva del usuario. Cada ítem se califica en una escala de Likert de 5 puntos. Esta escala es ampliamente utilizada por su simplicidad y confiabilidad estadística. [57]
- **Taller práctico basado en ISO/IEC 25022:** Tareas prácticas alineadas con las características y medidas de calidad en uso definidas en el modelo de calidad en uso, seccionada por objetivos para la evaluación del módulo de notificaciones.

El instrumento fue diseñado para ser aplicado a estudiantes usuarios de la aplicación móvil, quienes interactúan directamente con el sistema de notificaciones. La recopilación de sus percepciones permitirá identificar fortalezas y oportunidades de mejora en el sistema implementado.

A continuación, se detallan los instrumentos diseñados en las **Tablas 32 y 33**.

Tabla 32 Preguntas de encuesta SUS

N.º	Pregunta
1	¿Considera que utilizaría esta aplicación con frecuencia?
2	¿Considera que la aplicación es compleja?
3	¿Considera que la aplicación es fácil de utilizar?
4	¿Considera que necesitaría apoyo de un experto para poder utilizar esta aplicación?
5	¿Considera que la aplicación presenta inconsistencias?

6	¿Considera que esta aplicación es fácil de aprender para la mayoría de las personas?
7	¿Considera que la aplicación resulta confusa/difícil de utilizar?
8	¿Considera que las funciones de la aplicación están bien organizadas e integradas?
9	¿Se sintió seguro/a al utilizar la aplicación?
10	¿Considera que se requieren conocimientos técnicos previos para utilizar correctamente la aplicación?

Tabla 33 Taller práctico basado en ISO/IEC 25022

N.º	Objetivo	Tarea práctica
1	Visualizar el buzón de notificaciones	Ver el número de notificaciones recibidas en el ícono "campana" e ingresar al buzón
		Ver la sección de notificaciones leídas y sección de no leídas
		Seleccionar una notificación para redireccionarme
2	Seleccionar los tipos de notificaciones que deseo recibir	Seleccionar el módulo de notificaciones o ícono "campana"
		Navegar hasta la sección de recordatorios
		Activar o desactivar las notificaciones que deseo recibir
3	Marcar notificaciones como leídas	Ingresar a la sección del buzón de notificaciones
		En la sección de "no leídas" marcar la primera notificación como leída
		En la sección de "no leídas" seleccionar la opción "marcar todas como leídas"

		Ingresar a la sección del buzón de notificaciones "leídas"
4	Eliminar notificaciones del buzón	En la sección de "leídas" eliminar la primera notificación
		En la sección de "leídas" seleccionar la opción "eliminar todas"
5	Revisar el número de notificaciones sin leer	Regresar al "home" de la aplicación UTN-MÓVIL
		Visualizar si existen notificaciones pendientes en el ícono "campana"

3.4 Fiabilidad de los instrumentos

3.4.1 Fiabilidad del taller práctico

En la **Figura 41**, para evaluar la fiabilidad del taller práctico una vez obtenido los datos de los usuarios, se aplicaron las pruebas de normalidad de Kolmogorov-Smirnov y Shapiro-Wilk a las variables "Errores", "Tareas logradas" y "Objetivos logrados".

Pruebas de normalidad						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
→ Errores	,416	47	<.001	,575	47	<.001
Tareas logradas	,469	47	<.001	,535	47	<.001
Objetivos logrados	,460	47	<.001	,552	47	<.001

a. Corrección de significación de Lilliefors

Figura 41 Prueba de normalidad taller práctico

Los resultados mostraron que ninguna de las variables sigue una distribución normal ($p < .001$ en todas las pruebas), lo que justifica el uso de métodos no paramétricos para el análisis. A continuación, en la **Figura 42**, se calcularon correlaciones de Spearman para examinar las relaciones entre las variables.

			Errores	Tareas logradas	Objetivos logrados
Rho de Spearman	Errores	Coefficiente de correlación	1,000	-,652**	-,711**
		Sig. (bilateral)	.	<.001	<.001
		N	47	47	47
	Tareas logradas	Coefficiente de correlación	-,652**	1,000	,951**
		Sig. (bilateral)	<.001	.	<.001
		N	47	47	47
	Objetivos logrados	Coefficiente de correlación	-,711**	,951**	1,000
		Sig. (bilateral)	<.001	<.001	.
		N	47	47	47

** La correlación es significativa en el nivel 0,01 (bilateral).

Figura 42 Matriz de correlaciones taller práctico

Se encontró una correlación negativa significativa entre "Errores" y "Tareas logradas" ($\rho = -.652, p < .001$) y entre "Errores" y "Objetivos logrados" ($\rho = -.711, p < .001$), lo que indica que a medida que disminuyen los errores, aumentan tanto las tareas como los objetivos alcanzados. Además, se observó una correlación positiva fuerte y significativa entre "Tareas logradas" y "Objetivos logrados" ($\rho = .951, p < .001$), respaldando la coherencia interna del taller.

3.4.2 Fiabilidad de la encuesta SUS

De igual manera, una vez tabulados los datos de los encuestados, en la **Figura 43** se aplicó el método estadístico de alfa de Cronbach con el fin de evaluar la fiabilidad interna del cuestionario SUS. Previamente, se invirtieron los valores de las preguntas Q2, Q4, Q5, Q7 y Q10, las cuales correspondían al orden de los ítems redactados de forma negativa en la encuesta aplicada.

Estadísticas de fiabilidad	
Alfa de Cronbach	N de elementos
,819	10

Figura 43 Alfa de Cronbach encuesta SUS

El coeficiente obtenido fue de 0.819, lo que indica una buena consistencia interna entre los ítems del cuestionario, y respalda su uso como instrumento fiable para evaluar la percepción de usabilidad del sistema.

3.5 Evaluación del modelo de calidad en uso

Para evaluar las 3 características definidas en la **Tabla 31** se tomaron en cuenta los datos obtenidos del taller práctico ANEXO C y los datos tabulados de la encuesta ANEXO D.

3.5.1 Evaluación de eficacia

- **Tareas completadas:**

Para obtener esta métrica se utilizó la siguiente fórmula:

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

donde A representa el número de tareas completadas y B el número total de tareas asignadas. Al sustituir los valores $A = 596$ (tareas completadas por todos los usuarios) y $B = 658$ (total de tareas asignadas), se obtiene un resultado de 0.905, lo que indica que el 90.5% de las tareas fueron completadas exitosamente.

- **Objetivos logrados:**

Para calcular esta métrica se utilizó la siguiente fórmula:

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

donde A corresponde al número de objetivos alcanzados y B al total de objetivos definidos. Al reemplazar los valores ($A = 205$ y $B = 235$), se obtiene un resultado de 0.872, lo cual refleja que el 87.2% de los objetivos fueron alcanzados con éxito.

- **Errores en una tarea:**

Esta métrica se calculó con la siguiente fórmula:

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

donde A representa la cantidad de errores registrados en tareas completadas y B el número total de tareas. Al sustituir los valores ($A = 32$ y $B = 596$), se obtiene un resultado de 0.946, lo que indica que el 94.6% de las tareas se realizaron sin errores.

3.5.2 Evaluación de eficiencia

- **Tiempo de tareas:**

Esta métrica se calculó con la siguiente fórmula:

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

donde A corresponde al tiempo total utilizado por los usuarios expertos para completar las tareas, y B al tiempo total empleado por los usuarios nuevos. Al sustituir los valores, se obtiene un resultado de 0.831, lo que indica que los usuarios expertos completaron las tareas en un 83.1% del tiempo utilizado por los usuarios nuevos, reflejando una mayor eficiencia en la ejecución de las tareas.

- **Eficiencia del tiempo:**

Esta métrica fue calculada con la fórmula:

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

donde A representa el tiempo promedio utilizado por los usuarios expertos para completar los objetivos, y B el tiempo promedio empleado por los usuarios nuevos. Al aplicar los valores correspondientes, se obtuvo un resultado de 0.820, lo que indica que los usuarios expertos requirieron únicamente el 82% del tiempo que necesitaron los usuarios nuevos, evidenciando una mayor eficiencia en la ejecución de las tareas por objetivo.

3.5.3 Evaluación de satisfacción

- **Utilidad**

Para el cálculo de esta métrica, se asignaron valores numéricos específicos a las respuestas de la escala de Likert del cuestionario SUS, de acuerdo con la **Tabla 34**.

Tabla 34 Peso de respuestas SUS

ESCALA DE LIKERT	RESPUESTA	PESO
1	Totalmente en desacuerdo	0.2

2	En desacuerdo	0.4
3	Neutral	0.6
4	De acuerdo	0.8
5	Totalmente de acuerdo	1

Para la **Tabla 35**, se tomó en cuenta que los resultados de la pregunta P5 están invertidos.

Tabla 35 Resultado SUS - Utilidad

PREGUNTA	1	2	3	4	5	RESULTADO
P1	1	3	6	15	22	39 de 47
P5	5	5	9	16	12	33.2 de 47
P9	1	2	6	6	32	41.4 de 47
PROMEDIO						37.86

Con estos resultados, se procedió a aplicar la siguiente fórmula:

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

donde A representa el número de usuarios que consideran útil el aplicativo, y B el total de usuarios encuestados. Al sustituir los valores (A = 37.86 y B = 47), se obtiene un resultado de 0.805, lo que indica que aproximadamente el 80.5% de los usuarios perciben el aplicativo como útil.

- **Confianza:**

Para esta métrica se aplicó la siguiente fórmula:

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

donde A representa el total de usuarios que presentan reclamos y B el total de usuarios encuestados. Al remplazar los valores (A = 17 y B = 47), se obtuvo un resultado de 0.638, lo que representa una confianza del 63.8% de los usuarios.

- **Comodidad:**

Para el cálculo de esta métrica, se usaron los pesos asignados a las respuestas según la **Tabla 36** y se invirtieron previamente las repuestas a las preguntas P2, P4, P7 y P10:

Tabla 36 Resultados métrica de Comodidad

PREGUNTA	1	2	3	4	5	RESULTADO
P2	5	2	8	8	24	37 de 47
P3	0	2	4	10	31	42.2 de 47
P4	3	1	5	4	34	41.2 de 47
P6	0	2	3	13	29	42 de 47
P7	2	2	5	9	29	40.4 de 47
P8	3	2	6	5	31	40 de 47
P10	3	3	5	9	27	39 de 47
PROMEDIO						40.25

Una vez obtenidos los resultados, se aplicó la siguiente fórmula:

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

donde A representa al promedio de los usuarios que perciben el aplicativo cómodo y B el total de usuarios encuestados. Al remplazar los valores (A = 40.25 y B = 47), se obtuvo un resultado de 0.856, lo que representa un nivel de comodidad del 85.6% por parte de los usuarios encuestados.

3.6 Resultados del modelo de calidad en uso

La **Tabla 37** presenta un resumen integral de los resultados obtenidos para cada una de las métricas de calidad en uso evaluadas, correspondientes al modelo detallado previamente en la **Tabla 31**.

Tabla 37 Resultados del Modelo de Calidad en Uso

<i>Característica</i>	<i>Medida</i>	<i>Peso característica</i>	<i>Peso medida</i>	<i>Medición</i>	<i>Resultado</i>	<i>Resultado Característica</i>
Eficacia	Tareas completadas		16%	0.905	14.48%	37.89%
	Objetivos logrados	42%	16%	0.872	13.95%	
	Tareas sin errores		10%	0.946	9.46%	
Eficiencia	Tiempo de tareas	32%	16%	0.831	13.3%	26.46%
	Eficiencia del tiempo		16%	0.820	13.12%	

Satisfacción	Utilidad percibida		10%	0.805	8.05%	
	Confianza	26%	8%	0.638	5.1%	20%
	Comodidad		8%	0.856	6.85%	
TOTAL						84.35%

De acuerdo con los resultados, el sistema alcanzó un 84.35% en calidad de uso, cumpliendo con los parámetros de satisfacción definidos en la norma ISO/IEC 25022, como se visualiza en la **Figura 44**.

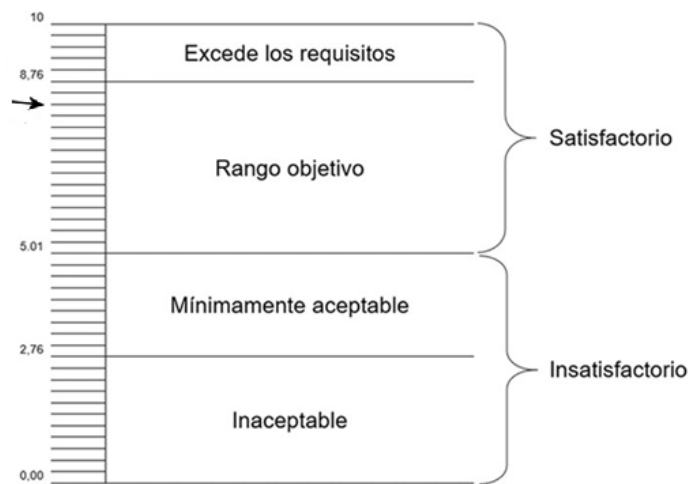


Figura 44 Resultado de calidad en uso [58]

CONCLUSIONES

El desarrollo del marco conceptual resultó fundamental para sentar las bases teóricas y técnicas que orientaron la implementación de un servicio de notificaciones push en la plataforma UTN-móvil. Este proceso permitió comprender los elementos clave relacionados con la gestión de la comunicación en el entorno académico tanto de docentes como estudiantes, así como seleccionar enfoques metodológicos y tecnológicos pertinentes.

La implementación del servicio “Notificaciones y Eventos” en la plataforma UTN-Móvil, permitió establecer un canal de comunicación más directo y eficiente entre la universidad y sus usuarios. Esta solución tecnológica facilitó el envío de información relevante a través de notificaciones a los usuarios finales de la aplicación móvil, lo que contribuyó a mantener informada a la comunidad universitaria de manera oportuna. La estructura del sistema demostró ser funcional y adaptable a las necesidades planteadas desde el inicio del proyecto.

Además, se aplicaron criterios de calidad en uso basados en la norma ISO/IEC 25022, lo que permitió evaluar el sistema desde la experiencia del usuario. Los resultados obtenidos reflejaron que el servicio cumple con niveles adecuados de efectividad, eficiencia y satisfacción, obteniendo un porcentaje del 83.4% de calidad en uso, lo cual valida su utilidad dentro del entorno universitario.

En general, el sistema desarrollado responde a una necesidad concreta de la comunidad académica. Al facilitar la comunicación institucional y permitir que tanto docentes como estudiantes estén al tanto de sus actividades, se fortalece el vínculo entre los actores educativos y se promueve una participación más activa en la vida universitaria.

RECOMENDACIONES

Se recomienda establecer un proceso de evaluación continua que permita medir la calidad del servicio a lo largo del tiempo. Aplicar periódicamente los criterios de la norma ISO/IEC 25022 ayudaría a detectar posibles mejoras y a mantener una experiencia positiva para los usuarios.

Es preciso extender el servicio de notificaciones push hacia diferentes sistemas operativos y dispositivos que no dependen de los servicios de Google, como ocurre con las nuevas generaciones de dispositivos Huawei. Esta estrategia permitirá garantizar una mayor cobertura, independencia tecnológica y equidad en la experiencia de los usuarios.

Se sugiere contemplar la integración del servicio de notificaciones con los nuevos módulos que se incorporen a la aplicación, así como con otros sistemas académicos, administrativos o externos que pudieran beneficiarse de este canal de comunicación. De esta manera, se consolidará la interoperabilidad y se incrementará el valor del servicio dentro del ecosistema institucional.

A mediano y largo plazo, se aconseja evaluar la viabilidad de extender el servicio de notificaciones hacia plataformas adicionales, como aplicaciones de escritorio, asistentes virtuales o incluso entornos web. Esta expansión fortalecería la presencia multiplataforma, incrementaría el alcance de la solución y ofrecería un mayor valor agregado a los usuarios.

REFERENCIAS

- [1] S. Zein, N. Salleh, and J. Grundy, "Systematic reviews in mobile app software engineering: A tertiary study," *Inf Softw Technol*, vol. 164, p. 107323, Dec. 2023, doi: 10.1016/J.INFSOF.2023.107323.
- [2] A. Hameed Pitafi and A. Ali, "An empirical investigation on actual usage of educational app: Based on quality dimensions and mobile self-efficacy," *Heliyon*, 2023, doi: 10.1016/j.heliyon.2023.e19284.
- [3] C. A. Trejos-Gil, D. A. Landínez, S. P. Vega, V. H. Meriño, C. Y. Martínez, and L. M. Ortíz, "Aplicaciones móviles para gestión de procesos académicos en educación superior (AMAES)," *Revista Espacios*, vol. 41, no. 28, 2020, Accessed: Apr. 09, 2024. [Online]. Available: <https://www.revistaespacios.com>
- [4] P. M. Viruel, L. A. Valiente, R. Palacios, and J. J. Bedolla, "Desarrollo de una aplicación móvil (APP) para el control exitoso de actividades académicas," *FORO DE ESTUDIOS SOBRE GUERRERO*, vol. 6, no. 1, pp. 785–802, 2019, Accessed: Apr. 09, 2024. [Online]. Available: <https://revistafesgro.cocytieg.gob.mx/index.php/revista/article/view/554>
- [5] Objetivos de Desarrollo Sostenible (ODS), "Objetivo 9: Industria, innovación e infraestructura | Programa De Las Naciones Unidas Para El Desarrollo." Accessed: Apr. 29, 2024. [Online]. Available: <https://www.undp.org/es/sustainable-development-goals/industria-innovacion-infraestructura>
- [6] R. Pérez, "Implementación de una aplicación móvil para fortalecer la gestión de notificaciones e incidencias ciudadanas del GAD municipal de San Miguel de Ibarra, basado en la característica de adecuación funcional de la ISO/IEC 25010," Jul. 2023, Accessed: Apr. 25, 2024. [Online]. Available: <https://repositorio.utn.edu.ec/handle/123456789/14594>
- [7] P. E. Bautista Salazar, "Aplicación móvil usando el framework Flutter para el control de las tutorías académicas del sistema integrado de la Universidad Técnica de Ambato," 2022, Accessed: Apr. 23, 2024. [Online]. Available: <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/34855>
- [8] B. A. Motz, M. G. Mallon, and J. D. Quick, "Automated Educative Nudges to Reduce Missed Assignments in College," *IEEE Transactions on Learning Technologies*, vol. 14, no. 2, pp. 189–200, Apr. 2021, doi: 10.1109/TLT.2021.3064613.
- [9] Rokhmadi, A. Sofwan, and M. Somantri, "Smart School System with Single ID based on RFID Through NFC using FCM Notification," *Proceedings - 2021 4th International Conference on Computer and Informatics Engineering: IT-Based Digital Industrial Innovation for the Welfare of Society, IC2IE 2021*, pp. 485–490, 2021, doi: 10.1109/IC2IE53219.2021.9649272.
- [10] K. S. Lakshmi, C. L. P. S. Sudha, Y. V. Bharathi, M. Divyanjali, and M. Suneetha, "Android Application on e-notifier for College Events Management using Firebase," *Proceedings of the 5th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2021*, 2021, doi: 10.1109/ICECA52323.2021.9676084.

- [11] Universidad Técnica del Norte, “Reglamento Portafolio Universitario,” 2014, [Online]. Available: https://legislacion.utn.edu.ec/wp-content/uploads/2017/12/2.REGLAMENTO_PORTAFOLIO_UNIVERSITARIO-1.pdf
- [12] T. Li, Y. Fan, Y. Li, S. Tarkoma, and P. Hui, “Understanding the Long-Term Evolution of Mobile App Usage,” *IEEE Trans Mob Comput*, vol. 22, no. 2, pp. 1213–1230, Feb. 2023, doi: 10.1109/TMC.2021.3098664.
- [13] Y. P. Bohórquez Uparela and M. T. Maza Figueroa, “Dispositivos móviles : evolución y uso,” <http://biblioteca.utb.edu.co/notas/tesis/0063136.pdf>, 2012, Accessed: Jun. 10, 2024. [Online]. Available: <https://repositorio.utb.edu.co/handle/20.500.12585/2767>
- [14] C. Guevara-Vega, J. Hernández-Rojas, M. Botto-Tobar, I. García-Santillán, A. Basantes Andrade, and A. Quiña-Mera, “Automation of the Municipal Inspection Process in Ecuador Applying Mobile-D for Android,” *Advances in Intelligent Systems and Computing*, vol. 1066, pp. 155–166, 2020, doi: 10.1007/978-3-030-32022-5_15.
- [15] L. D. Turner, S. M. Allen, and R. M. Whitaker, “The influence of concurrent mobile notifications on individual responses,” *Int J Hum Comput Stud*, vol. 132, pp. 70–80, Dec. 2019, doi: 10.1016/J.IJHCS.2019.07.011.
- [16] P. Tiffany, A. A. Pinem, A. N. Hidayanto, and S. Kurnia, “Gain-loss framing: Comparing the push notification message to increase purchase intention in e-marketplace mobile application,” *IEEE Access*, vol. 8, pp. 182550–182563, 2020, doi: 10.1109/ACCESS.2020.3029112.
- [17] G. Blinowski, A. Ojdowska, and A. Przybyłek, “Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation,” *IEEE Access*, vol. 10, pp. 20357–20374, 2022, doi: 10.1109/ACCESS.2022.3152803.
- [18] D. A. B. Contreras, “Arquitectura de Microservicios,” *Tecnología Investigación y Academia*, vol. 6, no. 1, pp. 36–46, Mar. 2018, Accessed: Jun. 11, 2024. [Online]. Available: <https://revistas.udistrital.edu.co/index.php/tia/article/view/9687>
- [19] I. Oumoussa and R. Saidi, “Evolution of Microservices Identification in Monolith Decomposition: A Systematic Review,” *IEEE Access*, vol. 12, pp. 23389–23405, 2024, doi: 10.1109/ACCESS.2024.3365079.
- [20] A. Quiña-Mera, D. Flores Landeta, X. M. Rea-Peñañiel, and C. Guevara-Vega, “Quality Evaluation of a Spring Cloud Microservices Architecture Implementation,” *Smart Innovation, Systems and Technologies*, vol. 252, pp. 745–754, 2022, doi: 10.1007/978-981-16-4126-8_66.
- [21] T. Prasandy, Titan, D. F. Murad, and T. Darwis, “Migrating application from monolith to microservices,” *Proceedings of 2020 International Conference on Information Management and Technology, ICIMTech 2020*, pp. 726–731, Aug. 2020, doi: 10.1109/ICIMTECH50083.2020.9211252.
- [22] D. A. Chuman Lluen, “Desarrollo de una aplicación móvil basada en microservicios para agilizar el proceso de ventas en un minimarket,” *Repositorio Institucional - USS*, 2023, Accessed: Jun. 14, 2024. [Online]. Available: <http://repositorio.uss.edu.pe/handle/20.500.12802/12164>

- [23] K. Schwaber and J. Sutherland, "La Guía Definitiva de Scrum: Las Reglas del Juego," 2020. Accessed: Jun. 14, 2024. [Online]. Available: <https://repositorio.uvm.edu.ve/handle/123456789/59>
- [24] Z. Masood, R. Hoda, and K. Blincoe, "Real World Scrum A Grounded Theory of Variations in Practice," *IEEE Transactions on Software Engineering*, vol. 48, no. 5, pp. 1579–1591, May 2022, doi: 10.1109/TSE.2020.3025317.
- [25] G. Guerrero, A. Guevara, J. A. Quiña-Mera, C. P. Guevara-Vega, and I. García-Santillán, "Software Project Management Integrating CMMI-DEV and SCRUM," *Communications in Computer and Information Science*, vol. 1535 CCIS, pp. 538–551, 2022, doi: 10.1007/978-3-031-03884-6_39.
- [26] A. Kusdiyanto, Suhardi, and W. Muhamad, "Combining Services Computing Systems Engineering Framework and Scrum for Agile Development," *2022 International Conference on Information Technology Systems and Innovation, ICITSI 2022 - Proceedings*, pp. 12–18, 2022, doi: 10.1109/ICITSI56531.2022.9970965.
- [27] The Blokehead, *Scrum - ¡Guía definitiva de prácticas ágiles esenciales de Scrum! - The Blokehead*. 2016. Accessed: Jun. 14, 2024. [Online]. Available: https://books.google.com.ec/books?hl=es&lr=&id=T24eDQAAQBAJ&oi=fnd&pg=PT17&dq=fases+de+scrum&ots=KVmxmS0KhX&sig=RTRdOUV81JbeQr3uhlip7EDrJog&redir_esc=y#v=onepage&q=fases%20de%20scrum&f=false
- [28] L. Urbina, M. Abud, G. Camarena, G. Alor Hernández, and A. I. Sánchez García, "Mixing Scrum-PSP: Combinación de Scrum y PSP para mejorar la calidad del proceso de software," 2016, Accessed: Jun. 14, 2024. [Online]. Available: https://rcs.cic.ipn.mx/2016_126/Mixing%20Scrum-PSP_%20Combinacion%20de%20Scrum%20y%20PSP%20para%20mejorar%20la%20calidad%20del%20proceso%20de%20software.pdf
- [29] Y. Zhao, Y. Hu, and J. Gong, "Research on International Standardization of Software Quality and Software Testing," *Proceedings - 2021 IEEE/ACIS 21st International Fall Conference on Computer and Information Science, ICIS 2021-Fall*, pp. 56–62, 2021, doi: 10.1109/ICISFALL51598.2021.9627426.
- [30] P. A. Roa, C. Morales, and P. Gutiérrez, "Norma ISO/IEC 25000," *Tecnología Investigación y Academia*, vol. 3, no. 2, pp. 27–33, Dec. 2015, Accessed: Jun. 15, 2024. [Online]. Available: <https://geox.udistrital.edu.co/index.php/tia/article/view/8373>
- [31] ISO, "ISO/IEC 25022:2016(en), Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Measurement of quality in use." Accessed: Jun. 15, 2024. [Online]. Available: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25022:ed-1:v1:en>
- [32] W. R. Pradanita and 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.

- [33] S. P. Vega Molina, "Desarrollo de un modelo de calidad externa de software para mejorar la eficiencia y portabilidad del producto final en aplicaciones móviles en la Empresa ADS - Software en el 2020," ESPE - Universidad de las Fuerzas Armadas, 2021. Accessed: Jun. 15, 2024. [Online]. Available: <https://repositorio.espe.edu.ec/bitstream/21000/25121/1/T-ESPEL-MAS-0034.pdf>
- [34] A. Quiña-Mera, Z. M. Guitarra de la Cruz, and C. Guevara-Vega, "Efficiency study of GraphQL and REST Microservices in Docker containers: A computational experiment," *Data and Metadata*, vol. 4, pp. 199–199, Feb. 2025, doi: 10.56294/DM2025199.
- [35] A. Quiña-Mera, C. Guevara-Vega, J. Caiza, J. Mise, and P. Landeta, "REST, GraphQL, and GraphQL Wrapper APIs Evaluation. A Computational Laboratory Experiment," *Lecture Notes in Networks and Systems*, vol. 614 LNNS, pp. 397–407, 2023, doi: 10.1007/978-981-19-9331-2_34.
- [36] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering – A systematic literature review," *Inf Softw Technol*, vol. 51, no. 1, pp. 7–15, Jan. 2009, doi: 10.1016/J.INFSOF.2008.09.009.
- [37] C. Guevara-Vega, B. Bernardez, A. Duran, A. Quina-Mera, M. Cruz, and A. Ruiz-Cortes, "Empirical Strategies in Software Engineering Research: A Literature Survey," *Proceedings - 2021 2nd International Conference on Information Systems and Software Technologies, ICI2ST 2021*, pp. 120–127, Mar. 2021, doi: 10.1109/ICI2ST51859.2021.00025.
- [38] P. Landeta-Lopez and C. Guevara-Vega, "Computational Experiments in Computer Science Research: A Literature Survey," *IEEE Access*, vol. 12, pp. 132254–132270, 2024, doi: 10.1109/ACCESS.2024.3458808.
- [39] A. Sterling, "NodeJS and Angular Tools for JSON-LD," *Proceedings - 13th IEEE International Conference on Semantic Computing, ICSC 2019*, pp. 392–395, Mar. 2019, doi: 10.1109/ICOSC.2019.8665625.
- [40] L. Puciarelli, *Node JS - Vol. 1: Instalación - Arquitectura - node y npm*. 2020. Accessed: Jul. 10, 2024. [Online]. Available: https://books.google.com.ec/books?hl=es&lr=&id=GOfqDwAAQBAJ&oi=fnd&pg=PP1&dq=nodejs&ots=uKhUX8QQaM&sig=MXAxsZ5Yl3-rEO5OdX31PQVl9w0&redir_esc=y#v=onepage&q=nodejs&f=false
- [41] D. M. Irfan Rizq and P. Irving Vitra, "Development of Backend Server Based on REST API Architecture in E-Wallet Transfer System," 2024, Accessed: Jul. 10, 2024. [Online]. Available: <https://journal.uii.ac.id/journalsnati/article/view/31954/16256>
- [42] Z. Bartłomiej and B. Marcin, "Comparative analysis of Node.js frameworks," 2024, Accessed: Jul. 10, 2024. [Online]. Available: <https://ph.pollub.pl/index.php/jcsi/article/view/5364/4434>
- [43] S. Arkadiusz and S. Tomasz, "Oracle 19c, SQL Server 2019, Postgresql 12 and MySQL 8 database systems comparison," *Journal Computer Sciences Institute*, 2020, doi: <https://doi.org/10.35784/jcsi.2281>.

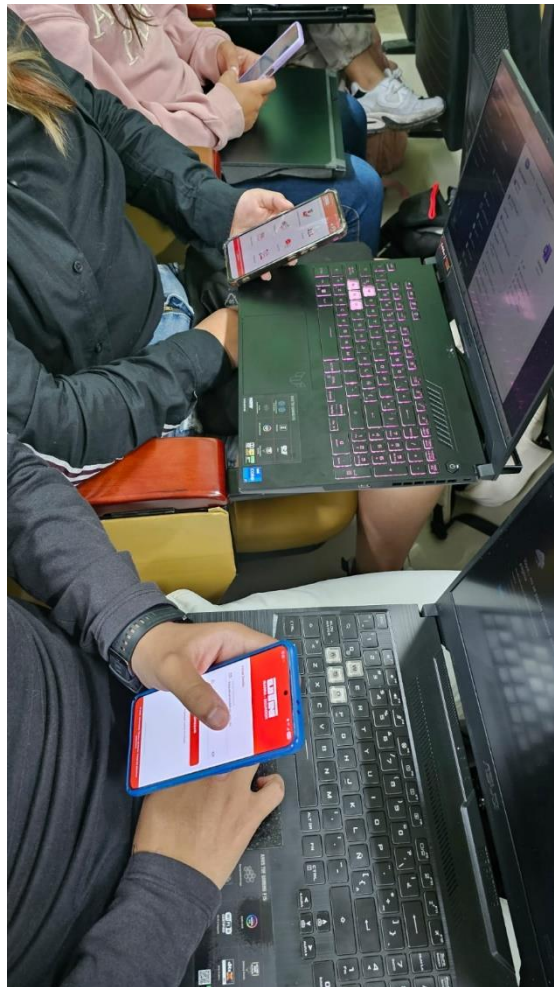
- [44] Steven. Feuerstein and Bill. Pribyl, "Oracle PL/SQL programming," p. 989, 2002.
- [45] J. Harish, S. Girish Rao, K. A. R. Ashok, Sharvani G.S, and D. Shivakumar, "React Apps with Server-Side Rendering: Next.js," *JOURNAL OF TELECOMMUNICATION, ELECTRONIC AND COMPUTER ENGINEERING (JTEC)*, 2022, doi: <https://doi.org/10.54554/jtec.2022.14.04.005>.
- [46] E. Windmill, *Flutter in Action - Eric Windmill - Google Libros*. 2020. Accessed: Jul. 09, 2024. [Online]. Available: https://books.google.com.ec/books?hl=es&lr=&id=EzgzEAAAQBAJ&oi=fnd&pg=PT19&dq=flutter&ots=7zpzF15qK3&sig=hGxfFQ8-qPKwocajl3C28vjkuM&redir_esc=y#v=onepage&q=flutter&f=false
- [47] K. Kishore, S. Khare, V. Uniyal, and 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.
- [48] M. Gusev, S. Ristov, G. Velkoski, and P. Gushev, "Alert notification as a service," *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2014 - Proceedings*, pp. 319–324, 2014, doi: 10.1109/MIPRO.2014.6859584.
- [49] S. K. Gudla, J. Bose, S. Sunkara, and S. Verma, "A unified Push Notifications service for mobile devices," *2015 IEEE International Conference on Electronics, Computing and Communication Technologies, CONECCT 2015*, Jan. 2016, doi: 10.1109/CONECCT.2015.7383922.
- [50] Y. Gao *et al.*, "Near real-time optimization of activity-based notifications," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 283–292, Jul. 2018, doi: 10.1145/3219819.3219880/SUPPL_FILE/GAO_ACTIVITY_BASED_NOTIFICATIONS.MP4.
- [51] L. Yali, "Las notificaciones push con estrategias de cambio de comportamiento como instrumento para aumentar el uso continuado de las APPS de fitness: un modelo de mediación moderada," UNIVERSIDAD COMPLUTENSE DE MADRID , Madrid, 2021. Accessed: Jul. 10, 2024. [Online]. Available: <https://docta.ucm.es/rest/api/core/bitstreams/626afa69-9e98-4de9-961b-f68ccd83fcd/content>
- [52] G. Albertengo, F. G. Debele, W. Hassan, and D. Stramandino, "On the performance of web services, google cloud messaging and firebase cloud messaging," *Digital Communications and Networks*, vol. 6, no. 1, pp. 31–37, Feb. 2020, doi: 10.1016/J.DCAN.2019.02.002.
- [53] L. Moroney, *Firestore Cloud Messaging*. Apress, Berkeley, CA, 2017. doi: 10.1007/978-1-4842-2943-9_9.
- [54] M. A. Faiz, D. S. Kusumo, and M. J. Alibasa, "Flutter Framework Code Portability Measurement on Multiplatform Applications with ISO 9126," *2022 1st International*

Conference on Software Engineering and Information Technology, ICoSEIT 2022, pp. 36–40, 2022, doi: 10.1109/ICOSEIT55604.2022.10030045.


- [55] J. M. Albero Belamendía, “Aplicación multiplataforma para la gestión de alumnos en academias,” 2023. Accessed: Jul. 10, 2024. [Online]. Available: <https://riunet.upv.es/handle/10251/198842>
- [56] N. E. JIMENEZ AVILA., “SISTEMA MÓVIL PARA NOTIFICACIONES DE LA UNIVERSIDAD TECNOLÓGICA ISRAEL,” 2021, Accessed: Jul. 09, 2024. [Online]. Available: <http://repositorio.uisrael.edu.ec/handle/47000/2774>
- [57] B. Klug, “An Overview of the System Usability Scale in Library Website and System Usability Testing,” *Weave: Journal of Library User Experience*, vol. 1, no. 6, Apr. 2017, doi: <https://doi.org/10.3998/weave.12535642.0001.602>.
- [58] “ISO/IEC 14598-1:1999 - Information technology — Software product evaluation — Part 1: General overview.” Accessed: Jul. 20, 2025. [Online]. Available: <https://www.iso.org/standard/24902.html>

ANEXOS

Anexo A. Fotografías del taller práctico y encuesta SUS a usuarios



Anexo B. Taller práctico de evaluación

		UNIVERSIDAD TÉCNICA DEL NORTE		FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS		CARRERA DE SOFTWARE			
		UTN 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 (Sí/No)								
4	Indique si completó la tarea con éxito (Sí/No)								
Nota:	Inicie cada objetivo desde la pantalla principal (Calendario de actividades)								
Nivel de carrera:									
Objetivo	Nro.	Descripción de la tarea	Inicio	Fin	Total	Error	Finalizado	Observación	
Visualizar el buzón de notificaciones	1	Ver el número de notificaciones recibidas en el ícono "campana" e ingresar al buzón			*****				
	2	Ver la sección de notificaciones leídas y sección de no leídas			*****				
	3	Seleccionar una notificación para redireccionarme			*****				
Seleccionar los tipos de notificaciones que deseo recibir	4	Seleccionar el módulo de notificaciones o ícono "campana"			*****				
	5	Navegar hasta la sección de recordatorios			*****				
	6	Activar o desactivar las notificaciones que deseo recibir			*****				
Marcar notificaciones como leídas	7	Ingresar a la sección del buzón de notificaciones			*****				
	8	En la sección de "no leídas" marcar la primera notificación como leída			*****				
	9	En la sección de "no leídas" seleccionar la opción "marcar todas como leídas"			*****				
Eliminar notificaciones del buzón	10	Ingresar a la sección del buzón de notificaciones "leídas"			*****				
	11	En la sección de "leídas" eliminar la primera notificación			*****				
	12	En la sección de "leídas" seleccionar la opción "eliminar todas"			*****				
Revisar el número de notificaciones sin leer	13	Regresar al "home" de la aplicación UTN-MÓVIL			*****				
	14	Visualizar si existen notificaciones pendientes en el ícono "campana"			*****				
					Total	0:00:00			
Final		Una vez finalizado el taller guárdelo y complete la encuesta							

Anexo C. Resultados del taller práctico

ID	ACULT	CARRE	NIVE	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	Errores	Áreas logra	Objetivos logra	Tiempo	Observación
1	FICA	Software	6	00:00:06	00:00:09	00:00:08	00:00:05	00:00:06	00:00:12	00:00:06	00:00:03	00:00:07	00:00:05	00:00:03	00:00:06	00:00:07	00:00:03	4	8	1	00:01:26	SI
2	FICA	Software	7	00:00:05	00:00:10	00:00:11	00:00:06	00:00:11	00:00:07	00:00:07	00:00:04	00:00:09	00:00:06	00:00:03	00:00:05	00:00:04	00:00:02	0	14	5	00:01:30	NO
3	FICA	Software	8	00:00:06	00:00:13	00:00:09	00:00:05	00:00:08	00:00:05	00:00:04	00:00:02	00:00:09	00:00:04	00:00:02	00:00:10	00:00:04	00:00:02	5	9	2	00:01:23	SI
4	FICA	Software	8	00:00:06	00:00:10	00:00:07	00:00:06	00:00:04	00:00:11	00:00:02	00:00:04	00:00:07	00:00:05	00:00:05	00:00:06	00:00:09	00:00:05	0	14	5	00:01:27	NO
5	FICA	Software	8	00:00:07	00:00:12	00:00:10	00:00:02	00:00:10	00:00:08	00:00:05	00:00:03	00:00:08	00:00:03	00:00:04	00:00:08	00:00:05	00:00:05	0	14	5	00:01:30	NO
6	FICA	Software	8	00:00:05	00:00:14	00:00:08	00:00:04	00:00:07	00:00:06	00:00:04	00:00:04	00:00:10	00:00:05	00:00:05	00:00:09	00:00:06	00:00:05	2	8	2	00:01:32	SI
7	FICA	Software	8	00:00:08	00:00:10	00:00:09	00:00:03	00:00:09	00:00:05	00:00:03	00:00:02	00:00:08	00:00:02	00:00:03	00:00:07	00:00:05	00:00:05	1	14	5	00:01:19	SI
8	FICA	Software	7	00:00:06	00:00:09	00:00:07	00:00:06	00:00:05	00:00:12	00:00:06	00:00:05	00:00:11	00:00:06	00:00:04	00:00:06	00:00:05	00:00:04	0	14	5	00:01:32	SI
9	FICA	Software	7	00:00:07	00:00:07	00:00:10	00:00:05	00:00:12	00:00:07	00:00:04	00:00:02	00:00:06	00:00:05	00:00:04	00:00:08	00:00:10	00:00:04	0	14	5	00:01:31	NO
10	FICA	Software	7	00:00:10	00:00:11	00:00:08	00:00:04	00:00:06	00:00:05	00:00:04	00:00:03	00:00:09	00:00:03	00:00:02	00:00:06	00:00:09	00:00:05	0	14	5	00:01:25	NO
11	FICA	Software	7	00:00:08	00:00:13	00:00:09	00:00:04	00:00:08	00:00:09	00:00:03	00:00:03	00:00:05	00:00:04	00:00:05	00:00:06	00:00:07	00:00:03	0	14	5	00:01:27	NO
12	FICA	Software	7	00:00:09	00:00:10	00:00:05	00:00:03	00:00:04	00:00:09	00:00:04	00:00:02	00:00:05	00:00:06	00:00:03	00:00:06	00:00:06	00:00:05	0	14	5	00:01:17	NO
13	FICA	Software	6	00:00:05	00:00:06	00:00:10	00:00:06	00:00:10	00:00:12	00:00:04	00:00:05	00:00:09	00:00:06	00:00:02	00:00:09	00:00:07	00:00:05	0	14	5	00:01:36	NO
14	FICA	Software	7	00:00:07	00:00:09	00:00:08	00:00:10	00:00:07	00:00:09	00:00:05	00:00:03	00:00:11	00:00:02	00:00:05	00:00:07	00:00:04	00:00:03	3	7	4	00:01:30	SI
15	FICA	Software	7	00:00:10	00:00:13	00:00:09	00:00:05	00:00:09	00:00:05	00:00:05	00:00:05	00:00:05	00:00:05	00:00:05	00:00:04	00:00:08	00:00:04	0	14	5	00:01:30	NO
16	FICA	Software	7	00:00:05	00:00:12	00:00:04	00:00:04	00:00:05	00:00:12	00:00:04	00:00:03	00:00:10	00:00:04	00:00:02	00:00:07	00:00:08	00:00:04	3	1	1	00:01:24	SI
17	FICA	Software	8	00:00:08	00:00:11	00:00:10	00:00:08	00:00:12	00:00:07	00:00:02	00:00:02	00:00:10	00:00:03	00:00:05	00:00:06	00:00:09	00:00:04	0	14	5	00:01:37	SI
18	FICA	Software	8	00:00:06	00:00:09	00:00:08	00:00:05	00:00:05	00:00:06	00:00:04	00:00:06	00:00:12	00:00:03	00:00:06	00:00:08	00:00:06	00:00:05	0	14	5	00:01:29	NO
19	FICA	Software	7	00:00:05	00:00:07	00:00:04	00:00:05	00:00:07	00:00:06	00:00:06	00:00:05	00:00:13	00:00:06	00:00:05	00:00:07	00:00:09	00:00:05	1	14	5	00:01:30	SI
20	FICA	Software	7	00:00:08	00:00:10	00:00:11	00:00:12	00:00:04	00:00:10	00:00:08	00:00:06	00:00:10	00:00:04	00:00:05	00:00:09	00:00:08	00:00:03	0	14	5	00:01:48	NO
21	FICA	Software	6	00:00:06	00:00:10	00:00:09	00:00:07	00:00:09	00:00:10	00:00:06	00:00:08	00:00:15	00:00:03	00:00:03	00:00:06	00:00:07	00:00:06	0	14	5	00:01:45	NO
22	FICA	Software	7	00:00:06	00:00:07	00:00:06	00:00:04	00:00:03	00:00:03	00:00:04	00:00:05	00:00:07	00:00:04	00:00:04	00:00:08	00:00:08	00:00:08	1	10	3	00:01:17	SI
23	FICA	Software	8	00:00:07	00:00:06	00:00:04	00:00:06	00:00:05	00:00:06	00:00:04	00:00:11	00:00:11	00:00:08	00:00:09	00:00:09	00:00:10	00:00:06	5	9	2	00:01:42	SI
24	FICA	Software	8	00:00:06	00:00:07	00:00:06	00:00:03	00:00:05	00:00:08	00:00:07	00:00:03	00:00:10	00:00:07	00:00:05	00:00:08	00:00:09	00:00:07	0	14	5	00:01:31	NO
25	FICA	Software	7	00:00:10	00:00:14	00:00:09	00:00:05	00:00:06	00:00:10	00:00:06	00:00:07	00:00:14	00:00:06	00:00:07	00:00:11	00:00:06	00:00:04	0	14	5	00:01:55	NO
26	FICA	Software	7	00:00:06	00:00:08	00:00:04	00:00:05	00:00:07	00:00:11	00:00:02	00:00:05	00:00:11	00:00:04	00:00:04	00:00:05	00:00:05	00:00:02	0	14	5	00:01:19	NO
27	FICA	Software	6	00:00:08	00:00:11	00:00:06	00:00:07	00:00:03	00:00:10	00:00:05	00:00:03	00:00:06	00:00:03	00:00:01	00:00:06	00:00:09	00:00:04	2	14	5	00:01:22	SI
28	FICA	Software	7	00:00:05	00:00:09	00:00:10	00:00:06	00:00:06	00:00:05	00:00:03	00:00:02	00:00:09	00:00:06	00:00:04	00:00:10	00:00:07	00:00:03	0	14	5	00:01:25	NO
29	FICA	Software	6	00:00:09	00:00:10	00:00:07	00:00:04	00:00:09	00:00:07	00:00:07	00:00:02	00:00:08	00:00:05	00:00:03	00:00:06	00:00:05	00:00:03	0	14	5	00:01:25	NO
30	FICA	Software	7	00:00:08	00:00:07	00:00:09	00:00:02	00:00:04	00:00:08	00:00:05	00:00:04	00:00:08	00:00:04	00:00:02	00:00:08	00:00:09	00:00:02	0	14	5	00:01:20	NO
31	FICA	Software	7	00:00:07	00:00:11	00:00:10	00:00:05	00:00:08	00:00:09	00:00:05	00:00:05	00:00:09	00:00:03	00:00:03	00:00:05	00:00:09	00:00:04	0	14	5	00:01:33	NO
32	FICA	Software	7	00:00:06	00:00:08	00:00:08	00:00:03	00:00:05	00:00:08	00:00:03	00:00:02	00:00:06	00:00:05	00:00:04	00:00:09	00:00:09	00:00:04	0	14	5	00:01:20	NO
33	FICA	Software	6	00:00:08	00:00:10	00:00:07	00:00:06	00:00:10	00:00:09	00:00:03	00:00:03	00:00:06	00:00:02	00:00:03	00:00:10	00:00:05	00:00:04	0	14	5	00:01:26	NO
34	FICA	Software	7	00:00:08	00:00:09	00:00:09	00:00:04	00:00:07	00:00:12	00:00:07	00:00:02	00:00:08	00:00:06	00:00:04	00:00:06	00:00:04	00:00:05	0	14	5	00:01:31	NO
35	FICA	Software	8	00:00:05	00:00:07	00:00:08	00:00:02	00:00:06	00:00:09	00:00:02	00:00:02	00:00:10	00:00:03	00:00:04	00:00:09	00:00:06	00:00:03	1	9	2	00:01:16	SI
36	FICA	Software	7	00:00:07	00:00:11	00:00:12	00:00:11	00:00:03	00:00:11	00:00:07	00:00:05	00:00:11	00:00:03	00:00:03	00:00:10	00:00:09	00:00:02	0	9	2	00:01:45	SI
37	FICA	Software	7	00:00:09	00:00:08	00:00:06	00:00:03	00:00:04	00:00:08	00:00:03	00:00:05	00:00:08	00:00:04	00:00:05	00:00:10	00:00:08	00:00:03	0	14	5	00:01:24	NO
38	FICA	Software	6	00:00:06	00:00:10	00:00:08	00:00:06	00:00:08	00:00:05	00:00:04	00:00:03	00:00:08	00:00:03	00:00:02	00:00:09	00:00:08	00:00:02	0	14	5	00:01:22	NO
39	FICA	Software	7	00:00:04	00:00:09	00:00:09	00:00:04	00:00:05	00:00:09	00:00:03	00:00:04	00:00:09	00:00:03	00:00:02	00:00:10	00:00:06	00:00:05	0	8	2	00:01:22	SI
40	FICA	Software	6	00:00:08	00:00:07	00:00:10	00:00:04	00:00:10	00:00:06	00:00:06	00:00:03	00:00:04	00:00:04	00:00:03	00:00:06	00:00:05	00:00:05	0	14	5	00:01:21	NO
41	FICA	Software	6	00:00:07	00:00:08	00:00:07	00:00:05	00:00:07	00:00:11	00:00:05	00:00:05	00:00:09	00:00:06	00:00:05	00:00:05	00:00:08	00:00:05	0	14	5	00:01:33	NO
42	FICA	Software	7	00:00:05	00:00:11	00:00:09	00:00:03	00:00:06	00:00:05	00:00:03	00:00:03	00:00:08	00:00:05	00:00:03	00:00:10	00:00:08	00:00:05	0	14	5	00:01:24	NO
43	FICA	Software	8	00:00:06	00:00:09	00:00:08	00:00:03	00:00:09	00:00:05	00:00:06	00:00:03	00:00:08	00:00:04	00:00:03	00:00:05	00:00:06	00:00:05	0	14	5	00:01:23	NO
44	FICA	Software	7	00:00:09	00:00:07	00:00:13	00:00:07	00:00:02	00:00:11	00:00:04	00:00:03	00:00:06	00:00:05	00:00:02	00:00:08	00:00:06	00:00:04	3	14	4	00:01:27	SI
45	FICA	Software	8	00:00:06	00:00:10	00:00:07	00:00:06	00:00:08	00:00:06	00:00:03	00:00:02	00:00:07	00:00:02	00:00:02	00:00:08	00:00:07	00:00:02	1	14	5	00:01:16	SI
46	FICA	Software	8	00:00:08	00:00:06	00:00:06	00:00:04	00:00:05	00:00:04	00:00:03	00:00:04	00:00:03	00:00:03	00:00:05	00:00:04	00:00:05	00:00:04	0	14	5	00:01:04	NO
47	FICA	Software	8	00:00:06	00:00:07	00:00:06	00:00:04	00:00:03	00:00:04	00:00:04	00:00:05	00:00:05	00:00:04	00:00:04	00:00:05	00:00:05	00:00:05	0	14	5	00:01:07	NO

Anexo D. Resultados encuesta SUS

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

27	FICA	Software	6	3	3	5	1	4	5	2	5	2	2
28	FICA	Software	7	4	1	4	1	1	2	1	5	2	1
29	FICA	Software	6	1	2	2	1	1	3	2	4	3	2
30	FICA	Software	7	4	2	5	1	2	4	2	4	3	1
31	FICA	Software	7	5	5	5	4	5	4	4	1	5	4
32	FICA	Software	7	4	5	5	1	3	5	1	5	5	1
33	FICA	Software	6	4	1	4	1	3	4	1	5	5	1
34	FICA	Software	7	4	2	4	2	2	4	2	4	4	2
35	FICA	Software	8	4	4	5	1	1	5	1	5	5	2
36	FICA	Software	7	2	3	3	2	2	4	1	3	3	2
37	FICA	Software	7	4	1	5	1	1	5	1	4	5	1
38	FICA	Software	6	4	3	4	1	3	5	3	3	4	1
39	FICA	Software	7	3	2	2	1	2	4	1	4	3	1
40	FICA	Software	6	3	2	4	1	3	5	1	5	5	1
41	FICA	Software	6	3	3	3	3	3	3	3	3	3	3
42	FICA	Software	7	5	3	4	3	4	4	3	3	3	2
43	FICA	Software	8	3	1	5	1	2	5	1	5	5	1
44	FICA	Software	7	2	1	3	2	3	4	4	2	4	5
45	FICA	Software	8	4	2	4	1	4	2	2	3	4	2
46	FICA	Software	8	5	1	5	1	1	4	1	5	5	1
47	FICA	Software	8	5	1	5	1	2	5	1	5	5	2