



Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

KUNAN: Aplicación Móvil para mejorar el proceso de registro de asistencia en la FISI

Curso Internet de las Cosas

AUTORES

Mitac Saavedra, Milena Diana	20200275
Rosas Sequeiros Fabricio	20200288
Salinas Mejías Ramsés Alfonzo	20200292

LIMA, PERÚ

2024

ÍNDICE GENERAL

1. CAPÍTULO 1: INTRODUCCIÓN	4
1.1. Situación Problemática	4
1.2. Justificación Teórica	4
1.3. Justificación Práctica	4
1.4. Objetivos	5
1.4.1. Objetivo General	5
1.4.2. Objetivos Específicos	5
1.5. Propuesta	5
2. CAPÍTULO 2: MARCO TEÓRICO	6
2.1. Antecedentes en el Ámbito Internacional	6
2.2. Antecedentes en el Ámbito Nacional	13
2.3. Bases Teóricas	14
3. CAPÍTULO 3: METODOLOGÍA	16
3.1. Desarrollo de la Metodología SCRUM	20
3.1.1. Fase de Iniciación	20
3.1.1.1. Visión del Proyecto	20
3.1.1.2. Lista de Requerimientos	20
3.1.1.3. Creación del Equipo SCRUM y definición de Roles	21
3.1.1.4. Product Backlog	23
3.1.1.5. Cronograma de Actividades	32
3.1.2. Fase de Planificación	33
3.1.2.1. Estimación de Historias de Usuario	33
3.1.2.1.1. Estimación con Story Points	33
3.1.2.1.2. Matriz de puntos de historia	34
3.1.2.2. Herramientas	34
3.1.2.3. Estimación de Costos	36
3.1.2.4. Prototipo	38
3.1.3. Fase de Implementación	43
3.1.3.1. Desarrollo de la Aplicación	45
3.2. Proceso de la prueba de usabilidad	47
3.2.1. Definir Objetivos	47
3.2.2. Identificar Usuarios Participantes	47
3.2.3. Desarrollar Tareas	48
3.2.4. Metodología propuesta	58
3.2.5. Preparar el Entorno de Prueba	59
3.2.5.1. Plataforma de Prueba:	59
4. CAPÍTULO 4: RESULTADOS Y DISCUSIÓN	65
5. REFERENCIAS	66
6. ANEXOS	67

1. CAPÍTULO 1: INTRODUCCIÓN

1.1. Situación Problemática

La toma de asistencia tradicional mediante llamadas individuales consume un tiempo considerable, lo que afecta la eficiencia del proceso educativo. En promedio, se pierde entre un 5% y un 10% del tiempo de clase en la toma de asistencia, lo que puede representar una pérdida significativa de horas de clase a lo largo de un semestre. Esta falta de automatización dificulta la precisión en el registro de la asistencia y genera obstáculos en la generación de informes. Además, la falta de aprovechamiento de tecnologías modernas disponibles limita la optimización de este proceso. Con el avance de las tecnologías de la información y la comunicación (TIC), existen múltiples herramientas que pueden ser utilizadas para mejorar la gestión de la asistencia. Sin embargo, muchas universidades aún no han adoptado estas soluciones. Se requiere una solución que sea fácil de implementar, accesible para estudiantes y profesores, y que garantice la seguridad y privacidad de los datos.

1.2. Justificación Teórica

La toma de asistencia en entornos universitarios es un proceso que, tradicionalmente, consume una cantidad significativa de tiempo debido a métodos manuales como llamadas individuales. Esta falta de automatización no solo afecta la eficiencia del proceso educativo, sino que también limita la precisión en el registro de la asistencia y dificulta la generación de informes. Además, la ausencia de aprovechamiento de tecnologías modernas disponibles agrava esta problemática.

1.3. Justificación Práctica

Automatizar el proceso de toma de asistencia permite recuperar el tiempo que se pierde actualmente en métodos manuales, mejorando así la eficiencia de las clases y permitiendo un mayor enfoque en el contenido educativo. La automatización facilita también la generación de informes detallados sobre la asistencia de los estudiantes, permitiendo a los profesores identificar patrones de asistencia y tomar decisiones informadas para mejorar la gestión educativa. Las herramientas tecnológicas modernas están diseñadas para ser intuitivas y

accesibles tanto para estudiantes como para profesores, integrándose sin problemas en el entorno educativo y minimizando la curva de aprendizaje. Asimismo, el uso de tecnologías avanzadas asegura que los datos de asistencia sean almacenados y manejados de manera segura, protegiendo la privacidad de los estudiantes y cumpliendo con las normativas legales. Finalmente, implementar una solución tecnológica para la toma de asistencia muestra un compromiso con la modernización y la innovación en la educación, mejorando la reputación de la institución y atrayendo a estudiantes y profesores que valoran un entorno educativo avanzado.

1.4. Objetivos

1.4.1. *Objetivo General*

- Desarrollar una aplicación móvil para la gestión de asistencia estudiantil en entornos universitarios.

1.4.2. *Objetivos Específicos*

- Optimizar el proceso de toma de asistencia en instituciones educativas mediante el uso de tecnología móvil.
- Mejorar la eficiencia del sistema educativo al reducir el tiempo dedicado a la toma de asistencia manual.
- Facilitar la generación de informes de asistencia de manera precisa y oportuna

1.5. Propuesta

Para abordar estas dificultades, se propone la implementación de una aplicación móvil que facilite la gestión de asistencia estudiantil mediante el uso de PIN y códigos QR dinámicos para asistencia remota; así como el uso de Bluetooth con el dispositivo de IoT ESP32-C3 para la asistencia presencial. Esta solución busca automatizar el proceso de toma de asistencia, haciendo que sea más eficiente y preciso, al tiempo que simplifica la generación de informes. La aplicación será desarrollada utilizando herramientas tecnológicas como Firebase para la autenticación de usuarios y gestión de notificaciones push, Flask como framework para el backend, Flutter para el desarrollo de la interfaz de usuario, y Figma para el diseño y prototipado de la interfaz.

2. CAPÍTULO 2: MARCO TEÓRICO

2.1. Antecedentes en el Ámbito Internacional

Antecedente 1: Location-aware Event Attendance System using QR Code and GPS Technology

El artículo presenta un sistema inteligente de control de asistencia a eventos universitarios que utiliza tecnologías QR y GPS. El sistema se compone de dos vistas principales: una vista de usuario para estudiantes y una vista de administrador para organizadores de eventos. Los estudiantes se autentican utilizando su correo electrónico y número de matrícula. Su asistencia se registra escaneando un código QR y su ubicación se rastrea mediante GPS. Esto garantiza que solo se registre la asistencia si el estudiante permanece en el evento hasta su finalización.

El sistema se ha probado exhaustivamente y ha demostrado ser preciso y confiable. Los usuarios también han expresado su satisfacción con la calidad de la interfaz, la confiabilidad del sistema y su facilidad de uso.

El sistema propuesto es una forma efectiva y eficiente de controlar la asistencia a eventos universitarios. Beneficia tanto a los estudiantes como a los organizadores de eventos al agilizar el proceso de registro de asistencia y garantizar una asistencia completa y precisa.

Antecedente 2: Automated Multi-factor Analytics for Cheat-Proofing Attendance-taking

Este paper propone un enfoque multifactorial para combatir el fraude en el registro de asistencia utilizando smartphones. El sistema se basa en la autenticación mediante códigos QR y códigos IMEI, junto con análisis estadístico de datos. Incluye una aplicación móvil para estudiantes y una aplicación web para profesores.

El sistema propuesto es más efectivo para detectar intentos de fraude en comparación con los métodos manuales tradicionales. Se compararon los registros de asistencia capturados por el sistema propuesto con los registros

manuales, y los resultados mostraron que el sistema propuesto era más efectivo para detectar intentos de fraude.

Antecedente 3: Design and Development of Attendance System Application Using Android-Based Flutter

El artículo "Design and Development of Attendance System Application Using Android-Based Flutter" describe una aplicación de sistema de asistencia para estudiantes desarrollada con Flutter en una plataforma Android. La aplicación aborda las ineficiencias y el fraude de los sistemas manuales de asistencia mediante la validación de asistencia con códigos QR y geolocalización.

El desarrollo incremental permite la iteración y mejora continuas. Se realizan pruebas exhaustivas, incluyendo la instalación, el uso, el escaneo de QR y la geolocalización. La evaluación de la opinión media del usuario (MOS) mide la utilidad y la calidad percibida. La aplicación funciona en varios modelos de Android y reduce significativamente el tiempo de asistencia. Se identifican áreas de mejora como la velocidad de carga de datos y la precisión de la geolocalización.

La aplicación ofrece una solución innovadora para mejorar la eficiencia y la integridad de la asistencia. Las tecnologías modernas como los códigos QR y la geolocalización proporcionan una forma más segura y conveniente de registrar la asistencia. El enfoque de desarrollo incremental permite adaptar la aplicación a las necesidades específicas de los usuarios. Las pruebas exhaustivas son esenciales para garantizar el funcionamiento adecuado en diferentes entornos y dispositivos. La evaluación de la opinión media del usuario proporciona información valiosa para futuras mejoras.

El estudio presenta una solución viable para los problemas de asistencia estudiantil, utilizando tecnologías modernas y un enfoque de desarrollo adaptable. Enfatiza la importancia de las pruebas rigurosas y la evaluación de la percepción del usuario para el éxito de la aplicación.

Antecedente 4: Smart-Hadir – Mobile Based Attendance Management System

Smart-Hadir es una aplicación móvil diseñada para gestionar la asistencia de estudiantes universitarios de forma digital. La solución utiliza tecnologías como NFC, códigos QR y balizas para registrar la asistencia de forma precisa y segura, superando las limitaciones de los métodos tradicionales.

Smart-Hadir ofrece ventajas como bajo costo y fácil implementación, integración fluida en la rutina diaria de profesores y estudiantes, funcionalidades adicionales como generación de informes y notificaciones, y mayor precisión y seguridad.

La efectividad de Smart-Hadir se evaluó a través de la retroalimentación de usuarios, tanto profesores como estudiantes. Los principales desafíos identificados fueron la resistencia al uso de smartphones por parte de algunos profesores y la falta de acceso a dispositivos móviles por parte de un pequeño número de estudiantes. Estos problemas se abordaron mediante el registro manual de asistencia cuando fuera necesario.

Antecedente 5: Integrating QR Code-Based Approach to University e-Class System for Managing Student Attendance

El artículo presenta un sistema automatizado basado en códigos QR para gestionar la asistencia de los estudiantes en un entorno universitario, integrándolo con el Sistema de Información Total del Campus (TCIS). En la era de la Revolución Industrial 4.0, que incluye análisis de grandes datos, inteligencia artificial, automatización y el Internet de las Cosas (IoT), el sistema e-Class pretende mejorar el proceso de captura de asistencia, reduciendo el tiempo perdido y los errores asociados con los métodos manuales.

El sistema desarrollado incluye una plataforma en línea para la Gestión Académica del Rendimiento Estudiantil (SMAP), el TCIS y el código QR como nueva interfaz para que los estudiantes registren su asistencia a través de la intranet de la universidad. Este sistema busca mejorar el registro de asistencia de los estudiantes, promover un registro sistemático y organizado y cumplir con el requisito de asistencia del 80% establecido por la universidad. Un impacto significativo del sistema e-Class ha sido el aumento de la tasa de

asistencia a un 96%, mostrando que los estudiantes están más conscientes de su asistencia.

Metodología: El desarrollo del sistema e-Class se enfoca en una función prototipo que captura, inserta, recupera y elimina información, escaneando los códigos QR generados automáticamente por el sistema TCIS a través de los teléfonos inteligentes de los estudiantes. Los estudiantes deben iniciar sesión en SMAP con sus teléfonos inteligentes y escanear el código QR para registrar su asistencia. Los componentes esenciales incluyen el código QR, que se captura con los teléfonos inteligentes y se decodifica mediante aplicaciones especializadas, registrando la información en la base de datos del servidor de la universidad.

Validación y Resultados: El sistema se probó durante dos semestres y demostró ser eficiente y fiable para el seguimiento y la generación de informes de asistencia. Los resultados mostraron que el sistema facilita la identificación y corrección de los estudiantes que no cumplen con la política mínima de asistencia. Los administradores y profesores pueden acceder a la información centralizada para gestionar a los estudiantes que no cumplen con los requisitos, emitiendo cartas de advertencia y alertas según sea necesario. Además, el sistema permite la generación de informes de asistencia, mejorando la monitorización y el cumplimiento del 80% de asistencia requerido por la universidad.

Antecedente 6: Timekeeping and Immediate Monitoring of Employees by Consistently Advocating Time Consciousness and Honesty Using Enhanced Attendance Monitoring System (TIME CATCH Using EAMS)

El artículo presenta el desarrollo de un sistema mejorado de monitoreo de asistencia utilizando un sistema de monitoreo de asistencia mejorada (EAMS, por sus siglas en inglés). La Isabela State University Angadanan Campus implementó un sistema biométrico para el monitoreo de asistencia, pero tenía limitaciones en cuanto al número de empleados registrados, la gestión de permisos y la generación de registros diarios de asistencia (DTR). Para solucionar estos problemas, se desarrolló el sistema EAMS, que utiliza tecnología de reconocimiento facial y lector de códigos QR para registrar la

asistencia electrónicamente y almacenar los registros en una base de datos. Este sistema permite almacenar múltiples empleados, procesar solicitudes de permisos, gestionar los créditos restantes de permisos y generar automáticamente los registros diarios de asistencia.

Durante la pandemia, se recomendó abandonar el uso de sistemas biométricos de huellas dactilares debido al riesgo de contagio por contacto. Como resultado, la universidad tuvo que recurrir temporalmente a libros de registro manuales, los cuales no garantizaban la seguridad de los empleados. El sistema biométrico existente también presentaba problemas de capacidad y confiabilidad. Para abordar estas cuestiones, se desarrolló el sistema TIME CATCH usando EAMS, que permite transacciones sin contacto y ofrece funcionalidades adicionales como el registro de empleados, generación automática de DTR, gestión de permisos y un panel de control en vivo para el monitoreo inmediato.

La metodología utilizada para el desarrollo del sistema incluyó el enfoque de Desarrollo Rápido de Aplicaciones (RAD). Se realizaron análisis y diseños rápidos, se contactó al personal relevante para determinar las características necesarias, y se recopilaron informes de recursos humanos para la automatización de reportes. El sistema fue diseñado para registrar la asistencia de los empleados mediante el escaneo de códigos QR y el reconocimiento facial, y para actualizar la asistencia en la base de datos tras la verificación. Se utilizó Visual Studio 2010 para desarrollar la aplicación de asistencia, PHP para la programación del módulo administrador y MySQL como motor de base de datos.

El sistema fue sometido a pruebas alfa y beta para detectar y solucionar problemas técnicos. La implementación incluyó el registro de empleados, donde se ingresaba la información personal y se imprimían tarjetas de código QR, la captura de asistencia mediante el escaneo de códigos QR y reconocimiento facial, y el monitoreo en vivo a través de un panel de control. Los empleados también podían solicitar permisos a través de un módulo específico y gestionar sus unidades de crédito mediante otro módulo. El sistema generaba reportes automáticamente, incluyendo la lista maestra de

empleados, formularios de solicitud de permisos y registros diarios de asistencia.

La evaluación del sistema se realizó mediante una encuesta basada en el estándar ISO 25010, dirigida a 28 empleados, tanto docentes como no docentes. Los resultados mostraron que el sistema fue altamente valorado en términos de funcionalidad, eficiencia, usabilidad, confiabilidad, mantenibilidad y portabilidad, con una media general de 4.72, indicando una fuerte aceptación por parte de los usuarios.

Antecedente 7: Teaching Management System with Applications of RFID and IoT Technology

Actualmente, las tecnologías IoT se aplican en diversas áreas como el transporte inteligente, ciudades inteligentes, hospitales y educación. La tecnología RFID es clave para implementar aplicaciones IoT y es comúnmente utilizada en universidades chinas como dispositivos de identificación de estudiantes. En este trabajo, se desarrolla un lector RFID compatible con WiFi (WiRF) utilizando plataformas de hardware de código abierto como Node MCU y RFID-RC522, que se usa para registrar automáticamente la asistencia de los estudiantes y su comportamiento en el aula. Además, los códigos QR se utilizan para acceder rápidamente a videos de cursos y realizar respuestas interactivas en tiempo real en el aula, fomentando el aprendizaje multidimensional y la motivación de los estudiantes.

La metodología del estudio incluye la implementación de un sistema de gestión educativa basado en IoT que integra terminales móviles, lectores RFID, tarjetas RFID, Node MCU y códigos QR. El sistema permite registrar la asistencia automáticamente cuando los estudiantes pasan sus tarjetas RFID al ingresar al aula y usar códigos QR para acceder a materiales de curso en video y ejercicios interactivos. También se registra el comportamiento en el aula mediante el lector RFID, permitiendo a los profesores registrar actitudes y participaciones específicas de los estudiantes.

El software del sistema incluye una aplicación Android para el registro y acceso de los estudiantes, funcionalidades para acceder a videos de curso mediante códigos QR, y el registro de asistencia basado en RFID. También se

implementa una función de respuesta interactiva en tiempo real mediante códigos QR y la capacidad de registrar el comportamiento en el aula en una base de datos en la nube.

La validación del sistema se llevó a cabo en un aula real, mostrando un aumento en la tasa de asistencia del 85% al 98% después de su implementación. Además, encuestas realizadas a 108 estudiantes y 6 profesores indicaron que el 96% de los estudiantes consideran que el sistema mejora el aprendizaje en el curso, y todos los profesores creen que mejora la calidad de la enseñanza en el aula.

Antecedente 8: Authenticating Physical Location Using QR Codes and Network Latency" proposes a novel mechanism to authenticate users

La autenticación se realiza mediante la transmisión de una secuencia de códigos QR desde un dispositivo emisor a un dispositivo escáner, verificando la validez de estos códigos y el tiempo de ida y vuelta de la transmisión para asegurar que el usuario se encuentra físicamente presente.

El sistema consta de tres componentes principales: un servicio de autenticación en la nube, un dispositivo emisor confiable y un dispositivo escáner del cliente. El proceso de autenticación implica la solicitud y recepción de códigos QR por el dispositivo emisor, que los muestra en pantalla. El dispositivo escáner captura estos códigos y los envía al servicio de autenticación, que verifica su validez y calcula el tiempo total de ida y vuelta. Si el escáner envía suficientes códigos correctos dentro del tiempo permitido, se concede la autenticación.

El estudio experimental se llevó a cabo utilizando una aplicación Android para los componentes de emisión y escaneo, y un servidor web NodeJS para el servicio de autenticación. Se evaluaron distintos factores que afectan los tiempos de ida y vuelta, como el rendimiento del dispositivo y el tipo de red. Los resultados mostraron que los tiempos de generación de los códigos QR varían según el dispositivo, pero que esta variación es pequeña en comparación con el impacto del tipo de red. Las pruebas de autenticación directa y mediante un colaborador remoto usando transmisión de video demostraron que los intentos de autenticación proxied (colaboradores remotos) son consistentemente más lentos que los intentos válidos (presencia física).

El análisis reveló que las conexiones remotas introducen un retraso significativo que puede ser detectado, permitiendo establecer un umbral de tiempo para diferenciar entre usuarios físicamente presentes y aquellos que intentan autenticar de manera remota. La configuración experimental incluyó diferentes dispositivos Android y tipos de red, mostrando que la latencia de red es un factor crítico y que el tiempo total de autenticación puede variar significativamente según el dispositivo y la red utilizados.

2.2. Antecedentes en el Ámbito Nacional

Antecedente 9: Desarrollo de una aplicación móvil para la registro y control de asistencia de la universidad estudiantes basados en Machine Learning

El artículo presenta la aplicación móvil "ASYS" diseñada para optimizar el registro y control de asistencias en una universidad peruana. La aplicación utiliza Firebase como gestor de datos y Machine Learning para la validación de asistencias mediante códigos QR.

El desarrollo se basó en la metodología de Programación Extrema (XP), incluyendo planificación, diseño, codificación, pruebas y lanzamiento. Se identificaron historias de usuario prioritarias y se recopilaron datos relevantes. En el diseño se crearon mockups y se estableció un modelo cliente-servidor. La codificación se realizó utilizando el patrón MVVM y Cloud Firestore para almacenar datos. Se integraron técnicas de Machine Learning para la validación y generación de respuestas automáticas. La aplicación fue evaluada mediante pruebas funcionales y encuestas de satisfacción.

La aplicación automatiza y mejora el registro de asistencias utilizando Firebase y Machine Learning. Simplifica la toma de asistencia con códigos QR, proporciona un sistema robusto y escalable para manejar grandes volúmenes de datos, e implementa respuestas automáticas a través de BrainShop.ai.

Se realizaron pruebas unitarias y de aceptación para asegurar el correcto funcionamiento de la aplicación. Se llevaron a cabo encuestas y presentaciones para medir la satisfacción de los usuarios. Los resultados mostraron un 100% de cumplimiento en funcionalidad y operabilidad, altos niveles de satisfacción (98%) y aceptación (92.9%). La Escala de Usabilidad del Sistema

(SUS) reveló que la mayoría de los usuarios consideraba la aplicación óptima, rápida y sencilla de usar. La aplicación "ASYS" representa una solución tecnológica moderna y eficiente para la gestión del registro de asistencias en entornos educativos. Su éxito demuestra el potencial de las tecnologías como Firebase y Machine Learning para abordar y resolver desafíos comunes en la educación. El modelo puede ser replicado en otras instituciones educativas.

2.3. Bases Teóricas

Aplicación Móvil

Una aplicación móvil es un programa o software diseñado para instalarse en dispositivos móviles electrónicos, ofreciendo una variedad de funciones y usos que abarcan televisión, teléfono, video, música, procesamiento de textos y acceso a Internet. (National Library of Medicine, 2024)

En el campo del desarrollo de software, hay tres tipos de aplicaciones: aplicaciones web, aplicaciones híbridas y aplicaciones nativas. A continuación, se detallan las distintas características de cada uno de estos tres tipos en la siguiente tabla.

Características	Aplicación web	Aplicación híbrida	Aplicación nativa
Uso	Los usuarios pueden acceder directamente desde un navegador	Los usuarios tienen que instalar la aplicación en un dispositivo	Los usuarios tienen que instalar la aplicación en un dispositivo
Rendimiento	Lento y con menos respuesta	Más rápido, pero puede consumir más batería	El rendimiento se puede optimizar según el dispositivo

Desarrollo	Rentable, con un plazo de comercialización más rápido	Rentable, con un plazo de comercialización más rápido	Caro, con un plazo de comercialización más lento
-------------------	-------------------------------------------------------	-------------------------------------------------------	--------------------------------------------------

Android

Android es un sistema operativo móvil diseñado para smartphones, tabletas y otros dispositivos con pantalla táctil. No obstante, también se utiliza en dispositivos como relojes inteligentes, televisores y sistemas multimedia de algunos automóviles. Desarrollado por Google, está basado en el Kernel de Linux, que es un núcleo de sistema operativo abierto y multiplataforma, lo que ha ayudado a popularizar numerosos dispositivos inteligentes al permitir el uso de una gran variedad de aplicaciones. (Adeva, 2023)

Metodología de desarrollo de Software

Es un conjunto de métodos y técnicas que se usan para diseñar un software de manera eficaz y eficiente. La elección de la metodología es fundamental para que los equipos tengan la capacidad de entregar proyectos a tiempo y dentro del presupuesto. Las metodologías de desarrollo de software se las pueden diferenciar por familias, entre ellas están: ágiles, tradicionales e híbridas. Las metodologías ágiles se basan en procesos iterativos y de corta duración, los métodos tradicionales se basan en un control de proceso, en donde se establece las actividades involucradas, el alcance del proyecto, los entregables a generar y las herramientas y seguimiento que se utilizarán, y por último las metodologías híbridas combinan las mejores prácticas de las metodologías tradicionales y ágiles, donde se reúne las ventajas de ambas (Cruz, 2018).

Singh, Bagga & Kaur (2020) afirman que existen diversos modelos para desarrollar un software que implementa un modelo predictivo, y entre ellos proponen las metodologías tradicionales y ágiles, de los cuales destacan los modelos: Cascada, Espiral, Scrum y XP.

Con ellos induce que se debe aplicar una de estas estrategias para que en el futuro el software a desarrollar sea adaptable y presente la menor cantidad de errores.

Scrum

Según “La Guía Definitiva de SCRUM”, Scrum es un marco de proceso utilizado para gestionar el desarrollo de productos complejos desde principios de la década de 1990, Scrum no es un proceso o tecnología para construir productos, es una estructura de proceso. Más bien, es un sistema en el que se pueden utilizar diferentes procesos y tecnologías. Scrum demuestra la relativa eficacia de las prácticas de gestión de productos y las prácticas de desarrollo para que podamos mejorar. El marco Scrum consta de equipos Scrum y sus roles, eventos, artefactos y reglas asociados. Cada elemento del marco tiene un propósito específico y es fundamental para el éxito de Scrum y su uso.

El Equipo Scrum

El equipo está formado por un Product Owner (Dueño de Producto), un Development Team (Equipo de Desarrollo) y un Scrum Master (Facilitador). Estos equipos siempre son autoorganizados y con capacidad de realizar varias tareas a la vez. Cada uno elige la forma en cómo desea llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. El modelo de equipo en Scrum está enfocado en optimizar la productividad, flexibilidad y creatividad.

3. CAPÍTULO 3: METODOLOGÍA

Las metodologías de desarrollo de aplicaciones o software tienen como principal objetivo producir un software de calidad, cumpliendo con los requerimientos del usuario final es decir se implementa varios procesos metodológicos que procuran mejorar la calidad del producto.

En función a la información recopilada por algunos investigadores se han detallado dos tipos de metodologías entre las más importantes.

Metodología Ágiles

Las metodologías ágiles tienen como función principal el desarrollo de software eficaz, ya que ayuda a tener una adaptación en el trabajo según las condiciones que presenta el aplicativo, contribuye con un trabajo de desarrollo flexible y evolutivo, también aporta un trabajo grupal con un buen sistema comunicativo y planificado.

Metodologías Clásicas

Las metodologías clásicas fueron elaboradas para tener un control ante el desarrollo de software cuando se comenzó a producir de forma masiva y es conocido como bien llamado proceso prescriptivo. Las metodologías que siguen un ciclo recurrente son:

- Cascada
- Incremental
- Prototipo
- Espiral

Diferencias entre las metodologías ágiles y clásicas

Metodología Ágiles	Metodologías Clásicas
Orientados a proyectos pequeños y de mediana magnitud que se realizan en tiempos cortos.	Orientados a proyectos de todas las magnitudes, en especial a los grandes proyectos.
El cliente forma parte del equipo de trabajo.	El cliente tiene interacción con el equipo de trabajo mediante reuniones
Disminuye el uso de las herramientas para obtener una mayor agilidad.	Aumenta el uso de las herramientas para la realización del trabajo.
Grupo de trabajo pequeño con buena comunicación.	Grupo de trabajo grande
Se realizan cambios que se presentan según el avance del proyecto.	Cualquier cambio que se presente debe ser notificado y constar en el contrato.

Metodología Scrum

En 1993, Jeff Sutherland realiza la adaptación como un esquema de desarrollo de aplicaciones o software (Ease), surge la metodología Scrum como un método orientado a los diferentes artículos tecnológicos.

El punto de mayor importancia de la metodología scrum son las reuniones, en específico las que se realizan después de una entrega ya que de esta manera se puede analizar e ir mejorando en el desarrollo del proyecto.

En el desarrollo de este proyecto, se trabajará con la metodología ágil Scrum; Scrum se centra en la entrega iterativa e incremental de un producto, priorizando la colaboración, la adaptabilidad y la transparencia.

En Scrum, el trabajo se organiza en sprints, que son períodos de tiempo fijos y cortos, típicamente de 1 a 4 semanas. Durante cada sprint, un equipo autoorganizado trabaja en las tareas más importantes, según lo determinado por el propietario del producto, quien representa las necesidades y requisitos del cliente o del negocio.

El proceso Scrum se basa en la inspección continua y la adaptación, lo que significa que el equipo revisa y ajusta su trabajo regularmente para mejorar la calidad y la eficiencia. Se utiliza un enfoque iterativo e incremental para maximizar el valor entregado al cliente en cada iteración.

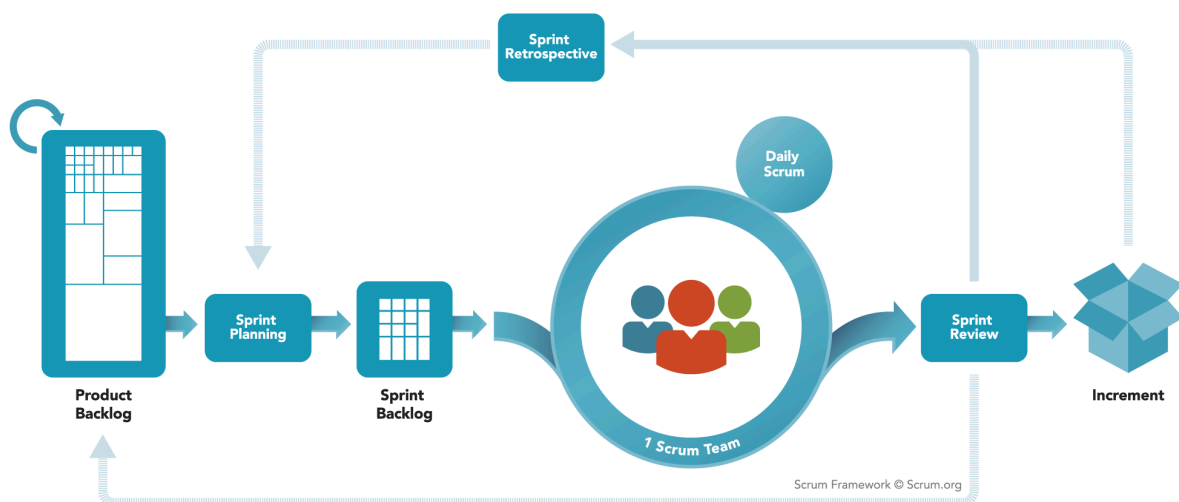


Figura 1: Proceso SCRUM

Roles de Scrum

- Product Owner: Es la persona indicada para la toma de decisiones, tiene conocimiento sobre el negocio junto con la visión de los diferentes productos, hace una lista de las ideas que dispone el cliente, las ordena según la prioridad y las ubica en el product backlog.
- ScrumMaster: Es el individuo que se encarga de la interacción con el cliente y de chequear que la metodología funcione de manera correcta en caso de la presencia de inconvenientes.
- Equipo de Desarrollo: Es un grupo considerado pequeño alrededor de 5 a 9 personas, que tienen un fin común de lograr un objetivo planteado.
- Usuarios: Es el individuo o un grupo de personas que será conocido como el cliente.
- Stakeholders: Es el grupo de personas que se beneficiarán por el proyecto, también tienen la función de revisar los Sprints.

Herramientas de Scrum

- Product Backlog: Listado de las necesidades que el cliente manifiesta.
- Sprint Backlog: Listado de las tareas de cada Sprint.
- Incremento: Es el fragmento funcional desarrollado que se añade al sprint.

Fases y Procesos

- Iniciación

El Product Owner se encarga de la definición de las necesidades del cliente, realizando un listado llamado como Product Backlog. Después de completar el paso anterior se procede a definir los roles de scrum master y el equipo de desarrollo.

- Planificación y Estimación

Se convoca a una reunión denominada Sprint planning meeting en la cual el Product Owner detalla las necesidades del cliente al equipo de desarrollo y al Scrum master.

El objetivo de la reunión es establecer las tareas y cumplir las necesidades del cliente las cuales se encuentran en el documento Sprint Backlog.

- Implementación

El proyecto de investigación se desarrollará según las diferentes tareas detalladas en periodos de tiempo establecidos llamados Sprints.

El punto más representativo en este paso son las reuniones diarias o Daily Scrum para constatar el avance del proyecto.

3.1. Desarrollo de la Metodología SCRUM

3.1.1. Fase de Iniciación

3.1.1.1. Visión del Proyecto

Transformar el proceso de toma de asistencia mediante la implementación de una solución tecnológica avanzada y eficiente. Esta solución busca eliminar la pérdida de tiempo asociada con diversos métodos, garantizar la precisión en el registro de datos, y facilitar la generación de informes detallados y útiles.

3.1.1.2. Lista de Requerimientos

ID-REQ	Requerimiento
REQ01	Permitir que los usuarios (estudiantes y profesores) inicien sesión de forma segura
REQ02	Implementar medidas de seguridad, como verificación de correo electrónico o autenticación de dos factores, para proteger la información.
REQ03	Incluir información como nombre, correo electrónico, foto de perfil, etc.
REQ04	Mostrar el horario de clases de los estudiantes y profesores.
REQ05	Permitir a los usuarios gestionar su horario y recibir notificaciones sobre próximas clases.

REQ06	Permitir a los profesores pasar asistencia a través de la aplicación, en clase presencial o remota
REQ07	Permitir a los estudiantes marcar su asistencias a través de la aplicación
REQ08	Permitir a los profesores determinar el periodo de toma de asistencia
REQ09	Implementar medidas de seguridad para evitar la suplantación de identidad en el paso de asistencias
REQ10	Entregar un reporte al docente al finalizar el periodo de toma de asistencia
REQ11	Alertar a los estudiantes y docentes cuando un curso se aproxime a iniciar
REQ12	Alertar a los estudiantes cuando un curso esté en pleno desarrollo y no se haya marcado su asistencia
REQ13	Alertar a los profesores cuando las horas límites de asistencia se aproximen

3.1.1.3. Product Backlog

Historia de Usuario	
Número: 1	Usuario: Profesor
Nombre de la historia: Iniciar sesión	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como profesor necesito tener acceso personalizado a la aplicación	
Validación:	

El sistema debe permitirle al profesor ingresar digitando su propio usuario y contraseña

Historia de Usuario	
Número: 2	Usuario: Profesor
Nombre de la historia: Crear usuario	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como profesor necesito crear un usuario y contraseña propio para tener un acceso personalizado a la aplicación	
Validación: El sistema debe permitirle al profesor crear un usuario nuevo y propio	

Historia de Usuario	
Número: 3	Usuario: Estudiante
Nombre de la historia: Iniciar sesión	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como estudiante necesito tener acceso personalizado a la aplicación	
Validación: El sistema debe permitirle al estudiante ingresar digitando su propio usuario y contraseña	

Historia de Usuario	
Número: 4	Usuario: Estudiante
Nombre de la historia: Crear usuario	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como estudiante necesito crear un usuario y contraseña propio para tener un acceso personalizado a la aplicación	
Validación: El sistema debe permitirle al estudiante crear un usuario nuevo y propio	

Historia de Usuario	
Número: 5	Usuario: Estudiante y profesor
Nombre de la historia: Protección de información	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como usuarios queremos contar con medidas de seguridad que verifiquen nuestra identidad para proteger nuestras cuentas	
Validación: El sistema valida con un método adicional a la contraseña la identidad del usuario	

Historia de Usuario	
Número: 6	Usuario: Estudiante y profesor
Nombre de la historia: Identificación de cuenta	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como usuarios queremos poder visualizar información con respecto a nuestro usuario al ingresar (nombre, correo y cualquier otra información de identificación)	
Validación: El sistema muestra información esencial del usuario al loguearse	

Historia de Usuario	
Número: 7	Usuario: Estudiante y profesor
Nombre de la historia: Clase actual	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como usuarios queremos poder visualizar la clase que nos toca según la hora real	
Validación: El sistema muestra la información correspondiente a la clase respectiva según la hora real	

Historia de Usuario	
Número: 8	Usuario: Estudiante y profesor
Nombre de la historia: Horario personal	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como usuarios queremos poder visualizar un horario personalizado con todas las clases que nos corresponden en la semana	
Validación: El sistema muestra un cronograma semanal con las clases correspondientes ordenadas por hora y día de la semana	

Historia de Usuario	
Número: 9	Usuario: Estudiante y profesor
Nombre de la historia: Modificación de horario	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como usuarios queremos poder agregar nuevos ítems dentro del horario que nos muestre la aplicación	
Validación: El sistema permite la creación de nuevos ítems dentro del horario y lo muestran en el mismo	

Historia de Usuario	
Número: 10	Usuario: Estudiante y profesor
Nombre de la historia: Notificación para aproximación de clase	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como usuario quiero recibir una recordatorio de la siguiente clase determinado tiempo antes de la misma, pudiendo decidir ese tiempo y la cantidad de veces que quiero recibirlo	
Validación: El sistema envía recordatorios de su próxima clase al usuario, cuándo y la cantidad de veces que lo especifique el mismo	

Historia de Usuario	
Número: 11	Usuario: Profesor
Nombre de la historia: Creación de asistencia	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como profesor requiero poder crear una solicitud de marcado de asistencia para los estudiantes en cualquier momento de la clase	
Validación: El sistema permite la creación de una solicitud de marcado de asistencia por parte del profesor para los estudiantes	

Historia de Usuario	
Número: 12	Usuario: Profesor
Nombre de la historia: Resultado de asistencia	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como profesor necesito acceder a una lista de asistentes e inasistentes culminar la asistencia	
Validación: El sistema le brinda al profesor una lista de los asistentes e inasistentes cada vez que culmina el pasado de asistencia	

Historia de Usuario	
Número: 13	Usuario: Profesor
Nombre de la historia: Asistencia para ambas modalidades de clase	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como profesor necesito que el pasado de asistencia se adapte a una clase presencial o virtual	
Validación: El sistema de pasado de asistencia se adapta según la modalidad de la clase	

Historia de Usuario

Número: 14	Usuario: Profesor
Nombre de la historia: Asistencia para clase presencial	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como profesor necesito que el sistema evite que estudiantes no presentes en el lugar de una clase presencial marquen asistencia	
Validación: El sistema valida que el estudiante que marca asistencia durante una clase presencial se encuentre en el salón	

Historia de Usuario	
Número: 15	Usuario: Estudiante
Nombre de la historia: Marcado de asistencia	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como estudiante necesito poder marcar mi asistencia una vez el profesor lo requiera	
Validación: El sistema permite que el estudiante marque la asistencia cuando el profesor lo requiera	

Historia de Usuario	
Número: 16	Usuario: Profesor

Nombre de la historia: Duración de la asistencia	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como profesor quiero poder determinar cuánto tiempo el estudiante accede a la asistencia	
Validación: El sistema permite que el profesor determina la duración de la solicitud de asistencia	

Historia de Usuario	
Número: 17	Usuario: Profesor
Nombre de la historia: Evitar suplantación de estudiante	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como profesor necesito que el sistema evite que un estudiante marque la asistencia de un compañero	
Validación: El sistema evita que un estudiante marque la asistencia de un compañero	

Historia de Usuario	
Número: 18	Usuario: Estudiante

Nombre de la historia: Notificación para pasado de asistencia	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como estudiante quiero recibir una notificación cuando el profesor abra el marcado de asistencia	
Validación: El sistema notifica al estudiante cuando el profesor abre el marcado de asistencia	

Historia de Usuario	
Número: 19	Usuario: Profesor
Nombre de la historia: Recordatorio para pasar asistencia	
Prioridad en negocio: --	Iteración asignada: --
Descripción: Como profesor necesito que el sistema me notifique 10 minutos antes de terminar la clase que debo pasar la asistencia si hasta el momento no lo hice	
Validación: El sistema notifica al profesor 10 minutos antes de terminar la clase que debe pasar la asistencia si hasta el momento no lo ha hecho	

3.1.1.4. Cronograma de Actividades

	Semanas
--	---------

[illegible]

3.1.2. Fase de Planificación

Durante esta fase, el equipo Scrum y el propietario del producto colaboran para determinar qué elementos del producto se trabajarán en el próximo sprint. Se crea un sprint backlog que contiene las tareas específicas que el equipo se compromete a completar durante el sprint. Se establece la meta del sprint, que describe el resultado deseado al final del sprint.

3.1.2.1. Estimación de Historias de Usuario

La estimación de historias de usuario es esencial en el desarrollo ágil de software, permitiendo asignar un nivel de esfuerzo relativo a cada tarea mediante técnicas de estimación de Historias de Usuario. Para el presente trabajo se ha escogido la técnica de Story Points.

3.1.2.1.1. Estimación con Story Points

La técnica de estimación con Story Points es fundamental en Scrum, utilizando una unidad relativa de medida para asignar esfuerzo a las historias de usuario. Esta unidad se basa en la comparación con tareas previamente estimadas o una historia de referencia.

A medida que avanza el proyecto, las estimaciones se vuelven más precisas gracias al mayor conocimiento adquirido. La técnica no busca determinar inicialmente el tiempo requerido para cada requisito, sino su dificultad, lo que facilita la priorización del Product Backlog y fomenta el entendimiento compartido sobre los requisitos.

3.1.2.1.2. Matriz de puntos de historia

La matriz de puntos historias sirve para aclarar y brindar una idea de como calificar cada tarea. Esta matriz evolucionará en el tiempo a medida que se ejecuten los sprints.

Punto de historia	Cantidad de esfuerzo requerido	Tiempo necesario	Complejidad de la tarea	Riesgo o incertidumbre de la tarea
1	Esfuerzo mínimo	Algunos minutos	Poca complejidad	Ninguno
2	Esfuerzo mínimo	Algunas horas	Poca complejidad	Ninguno
3	Esfuerzo leve	Un día	Baja	Baja

			complejidad	
5	Esfuerzo moderado	Algunos días	Complejidad media	Moderada
8	Esfuerzo elevado	Una semana	Complejidad media	Moderada
13	Esfuerzo máximo	Un mes	Alta complejidad	Alta

3.1.2.2. Herramientas

En el proceso de desarrollo de la aplicación móvil, se emplearán diversas herramientas tecnológicas para facilitar la implementación y el diseño de la aplicación. A continuación se detallan algunas de las principales herramientas que se utilizarán en este proyecto:

Tecnología	Descripción
Firestore	Se emplea la plataforma Firestore de Google para gestionar aspectos clave de la aplicación, como la autenticación de usuarios y la gestión de notificaciones push.
Flask	Para el desarrollo del backend de la aplicación, se utilizará el framework Flask de Python. Flask proporciona una estructura sólida para la creación de APIs y la lógica de negocio de la aplicación, permitiendo una comunicación eficiente entre el cliente y el servidor.
Flutter	Para la creación de la interfaz de usuario de la aplicación, se emplea el framework Flutter de Google. Flutter permite desarrollar interfaces de usuario nativas y altamente personalizables para dispositivos iOS y Android, utilizando un único código base en el lenguaje de

	programación Dart.
Modelado de Interfaz Figma	Para el diseño y prototipado de la interfaz de usuario, se utilizará la herramienta Figma. Figma proporciona una plataforma colaborativa en la nube para crear y compartir diseños de manera eficiente, facilitando la iteración y la retroalimentación del equipo de desarrollo y diseño.
GitHub	Se utilizará GitHub como plataforma de gestión de código fuente y colaboración. GitHub permite a los desarrolladores trabajar de manera colaborativa en el código, realizar seguimiento de los cambios y gestionar problemas y solicitudes de cambios de manera eficiente.
Jira	Se emplea Jira como herramienta de gestión de proyectos ágil. Jira permite a los equipos planificar, realizar seguimiento y gestionar tareas y sprints de desarrollo de manera eficiente, facilitando la colaboración y la comunicación dentro del equipo.
ESP32	Es un microcontrolador económico y eficiente en energía que integra un núcleo de CPU con capacidades de Wi-Fi y Bluetooth LE. Ofrece 400 KB de RAM, soporte para memoria Flash externa, y una variedad de periféricos programables. Diseñado para aplicaciones IoT, domótica, sensores y dispositivos portátiles.

- Diagrama de concepto:

Diagrama de concepto teniendo en cuenta el dispositivo de IoT a utilizar para la solución:

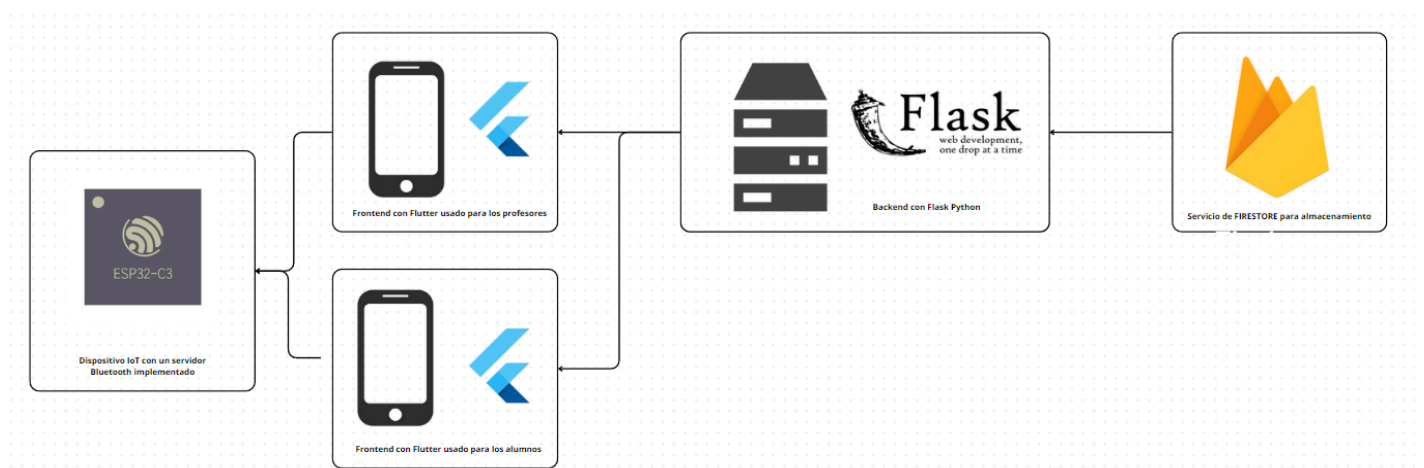


Figura 2: Diagrama de concepto

3.1.2.3. *Estimación de Costos*

- *Desarrollo y Diseño:*
 - Costo por hora de desarrollo y diseño: \$10 por hora.
 - *Backend:*
 - Costo por hora de desarrollo backend: \$15 por hora.
 - *Frontend:*
 - Costo por hora de desarrollo frontend: \$15 por hora .
- *Firebase:*
 - Autenticación de usuarios: \$35 mensuales.
 - Cloud Functions : \$20 mensuales.
- *Pruebas y Dispositivos:*
 - Pruebas en diferentes dispositivos móviles: \$250 mensuales
- *Licencias y Herramientas:*
 - *Figma:*
 - Costo por suscripción mensual de Figma: \$0 mensuales.
 - *GitHub:*
 - Costo por suscripción mensual de GitHub: \$0 mensuales.(plan gratuito para equipos pequeños)
 - *Jira*
 - Costo por suscripción mensual de Jira: \$0 mensuales.(plan gratuito para equipos pequeños)
- *Publicación y Mantenimiento:*
 - Google Play Store \$25 único
 - Apple App Store \$99 Anual
 - Mantenimiento: 20% del costo total de desarrollo
- *Dispositivos de IoT:*
 - Arduino \$25 único

- ESP32-C3 \$50 único
- Cables y adicionales \$15 único

TOTAL:

DETALLE	COSTO (Dólares)	TIEMPO	TOTAL (Dólares)
Diseño	10 x hora	80 Horas	800
Desarrollo	15 x hora	240 Horas	3,600
Firebase	55 x mes	2 meses	110
Licencias	0	-	0
Pruebas	250 x mes	1 mes	250
Publicación	124 x única vez	-	124
Dispositivos de IoT	\$90 x única vez		90
Mantenimiento	20% total	-	976.80
TOTAL	-	-	5,950.80

3.1.2.4. Prototipo

Pantalla de inicio:



Kunan

Inicio ►



Facultad de Ingeniería de
Sistemas e Informática

Pantalla de inicio de sesión:



Kunan

Correo

Ingresa tu correo

Contraseña

Ingresa tu contraseña

Iniciar Sesión

¿No tienes una cuenta? [Regístrate](#)

Pantalla principal:

¿Quién soy?



Estudiante



Profesor

Registro:



Kunan

Nombre de Usuario

Ingresar tu nombre de usuario

Correo

Ingresar tu correo

Contraseña

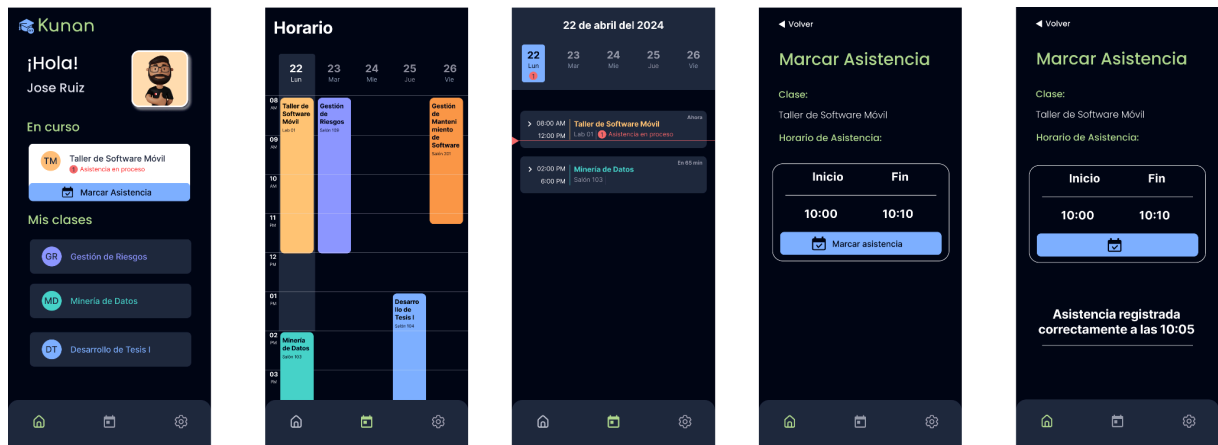
Ingresar tu contraseña

Confirmar Contraseña

Ingresar tu contraseña

Registrarme

Vista estudiante:



Vista Profesor:



3.1.3. Fase de Implementación

TallerMovil

Proyecto de software

PLANIFICACIÓN

Cronograma

Backlog

Tablero

Lista

Objetivos

Incidentes

Añadir vista

DESARROLLO

Código

Páginas de proy...

Añadir acceso rápido

Configuración del p...

Estás en un proyecto gestionado por el equipo

Más información

Proyectos / TallerMovil

Backlog

AF

Epic

Etiqueta

Tablero Sprint 1

Añadir fechas

(4 incidencias)

000

Iniciar sprint

Tablero Sprint 2

Añadir fechas

(5 incidencias)

000

Iniciar sprint

TM-16 HU-5 Protección de información

TAREAS POR H...

TM-17 HU- 6 Identificación de cuenta

TAREAS POR H...

TM-18 HU-7 Clase actual

TAREAS POR H...

TM-19 HU-8 Horario personal

TAREAS POR H...

TM-20 HU-9 Modificación de horario

TAREAS POR H...

+ Crear incidencia

5 incidencias | Estimación: 0

Tablero Sprint 3

Añadir fechas

(5 incidencias)

000

Iniciar sprint

Tablero Sprint 4

Añadir fechas

(5 incidencias)

000

Iniciar sprint

Backlog

(0 incidencias)

000

Crear sprint

Jira

Tu trabajo

Proyectos

Filtros

Paneles

Equipos

Aplicaciones

Crear

Mejorar

Busc

TallerMovil

Proyecto de software

PLANIFICACIÓN

Cronograma

Backlog

Tablero

Lista

Objetivos

Incidentes

Añadir vista

DESARROLLO

Código

Páginas de proy...

Añadir acceso rápido

Configuración del p...

Estás en un proyecto gestionado por el equipo

Más información

Proyectos / TallerMovil

Incidentes

Compartir

Exportar incidencias

Ir a todas las incidencias

VI

Buscar inci...

Proyecto: TallerMovil

Tipo

Estado

Persona asignada

Más +

Guardar fi

Creado

↓

↑

↺

↻

HU-19 Recordatorio para pasar asistencia

TM-30

HU-18 Notificación para pasado de asistencia

TM-29

HU-17 Evitar suplantación de estudiante

TM-28

HU-16 Duración de la asistencia

TM-27

HU-15 Marcado de asistencia

TM-26

HU-14 Asistencia para clase presencial

TM-25

1-19 de 19

Añadir epic / TM-30

HU-19 Recordatorio para pasar asistencia

Descripción

Usuario: Profesor

Descripción: Como profesor necesito que el sistema me notifique 10 minutos antes de terminar la clase que debo pasar la asistencia si hasta el momento no lo hice.

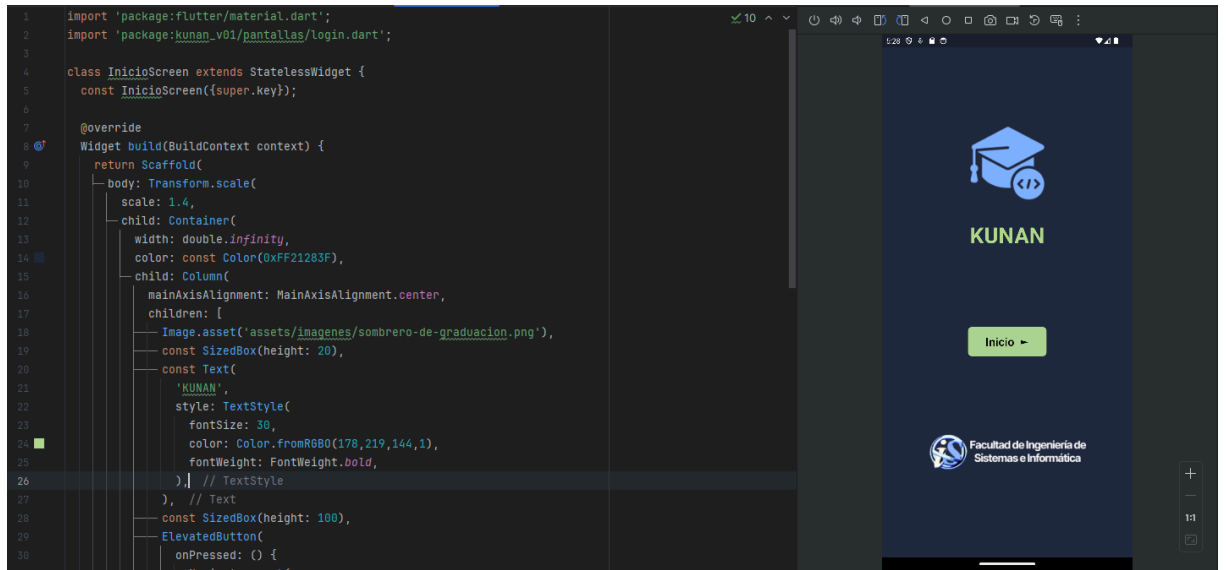
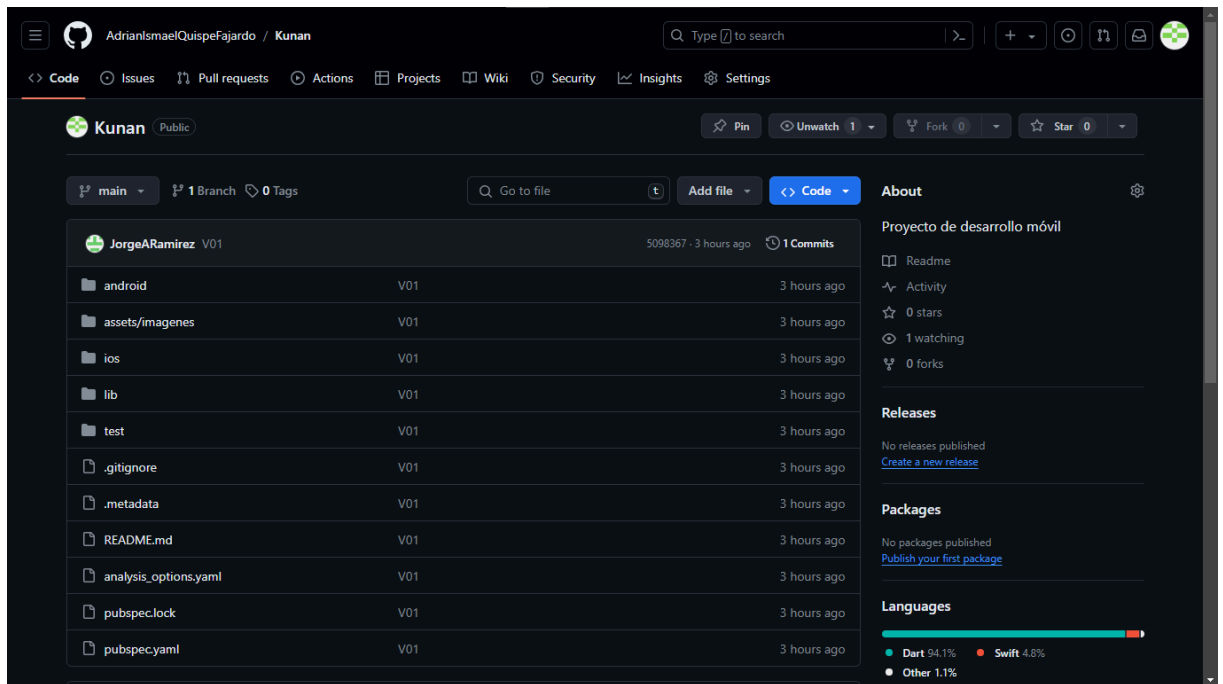
Validación: El sistema notifica al profesor 10 minutos antes de terminar la clase que debe pasar la asistencia si hasta el momento no lo ha hecho

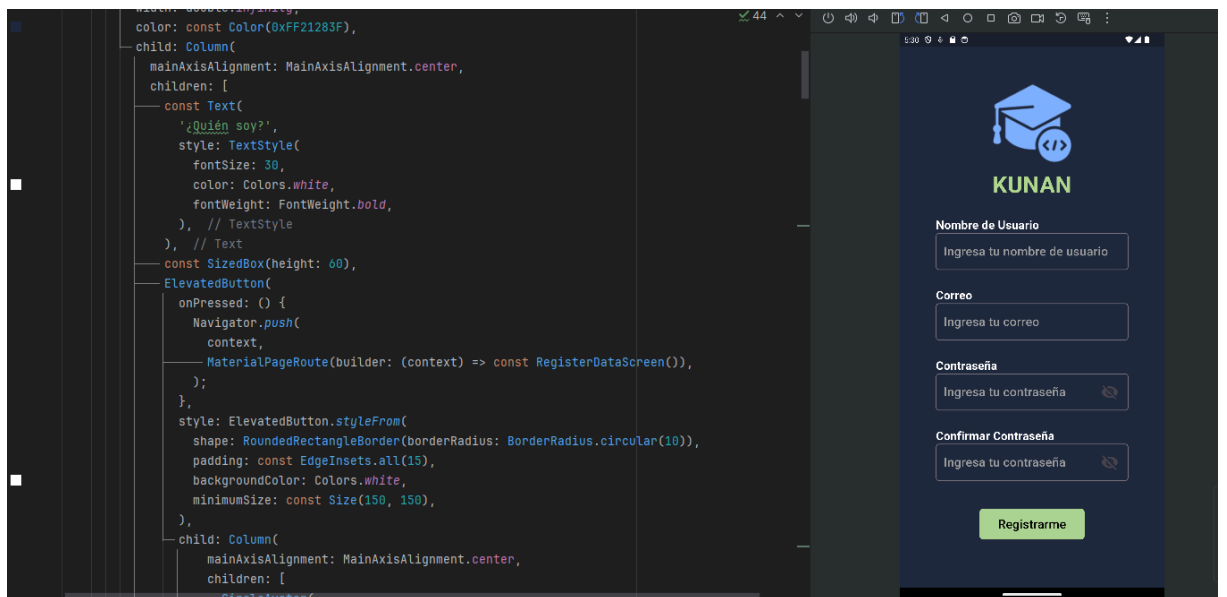
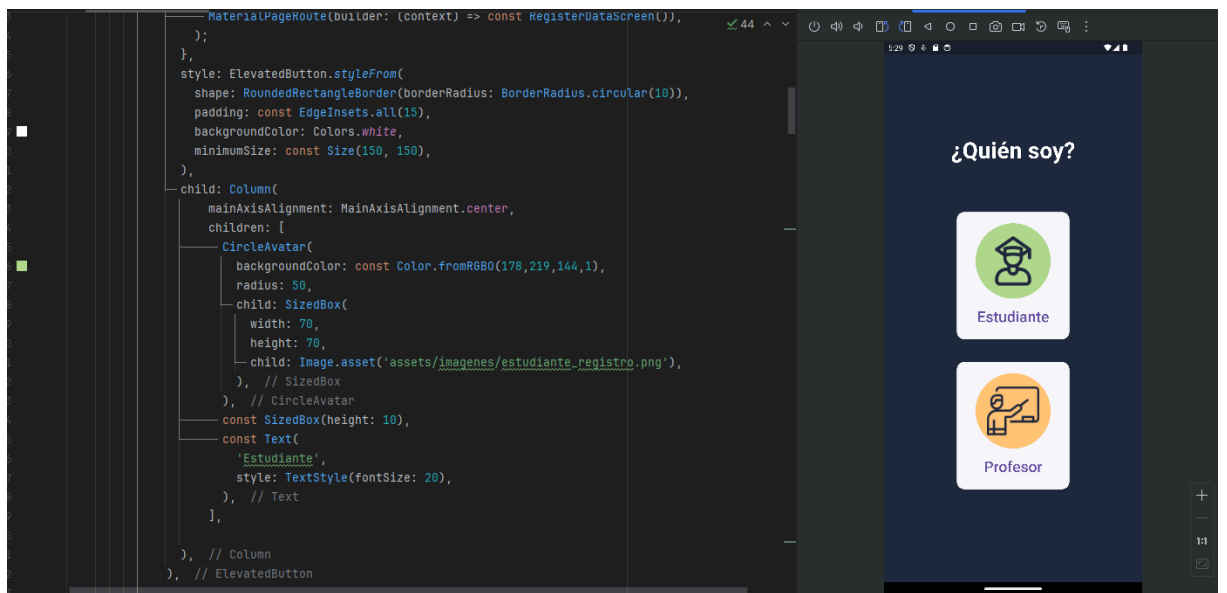
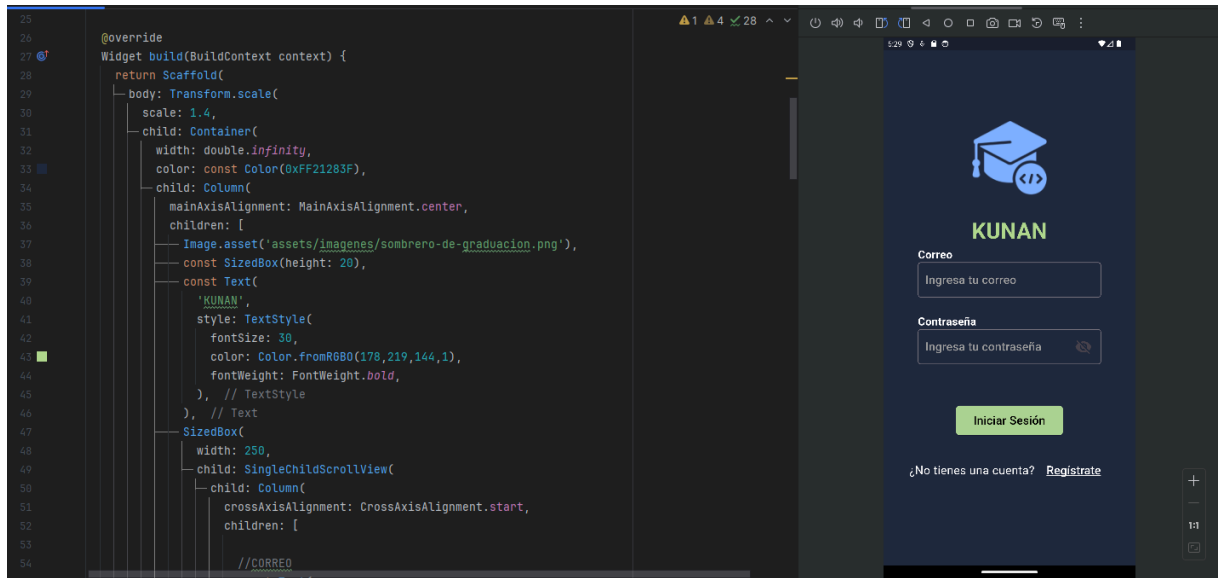
AF

Añadir un comentario...

Consejo de expertos: pulsa M para comentar

3.1.3.1. Desarrollo de la Aplicación





3.2. Proceso de la prueba de usabilidad

3.2.1. Definir Objetivos

3.2.1.1. Garantizar una Experiencia de Usuario Satisfactoria:

Asegurar que la aplicación sea fácil de usar y que los usuarios puedan cumplir sus tareas sin problemas ni confusiones.

3.2.1.2. Identificar y Corregir Problemas de Usabilidad:

Detectar cualquier obstáculo o problema que los usuarios puedan encontrar al interactuar con la aplicación y realizar los ajustes necesarios para solucionarlos.

3.2.1.3. Optimizar la Interfaz y Funcionalidades:

Evaluar cómo los usuarios interactúan con la interfaz y las funcionalidades de la aplicación para mejorar su diseño y rendimiento.

3.2.1.4. Validar el Diseño y la Accesibilidad:

Confirmar que la aplicación sea accesible para todos los usuarios, incluidos aquellos con discapacidades, y que funcione correctamente en diferentes dispositivos y contextos de uso.

3.2.2. Identificar Usuarios Participantes

a. Estudiantes de la Facultad de Ingeniería de Sistemas e Informática de la UNMSM

● Perfil:

- Edad promedio: 17-30 años
- Nivel: Pregrado-Posgrado
- Clases: Presenciales y Virtuales

● Justificación:

- Los estudiantes serán los principales usuarios de la aplicación, ya que serán quienes marquen su asistencia. Fundamental para comprender su experiencia y garantizar que la aplicación sea fácil de usar y accesible a todos.

b. Profesores de la Facultad de Ingeniería de Sistemas e Informática de la UNMSM

● Perfil:

- Edad promedio: 30-60 años

- Nivel: Profesionales con experiencia en enseñanza.
- Clases: Presenciales y Virtuales

- **Justificación:**

- Los profesores serán los encargados de verificar y gestionar las asistencias de los estudiantes. Ellos necesitan una interfaz que les permita monitorear las asistencias de manera eficiente.

3.2.3. Desarrollar Tareas

3.2.3.1. Tarea 1: Iniciar sesión en la aplicación

3.2.3.1.1. Descripción del contexto:

Imagina que eres un profesor que necesita acceder a la aplicación móvil de gestión de asistencia estudiantil.

3.2.3.1.2. Instrucciones específicas

- 3.2.3.1.2.1. *En la pantalla principal, identifica y selecciona la opción para iniciar sesión.*
- 3.2.3.1.2.2. *Ingresa tu nombre de usuario en el campo correspondiente.*
- 3.2.3.1.2.3. *Si las credenciales son correctas, verifica que el sistema te permite acceder a tu cuenta.*

3.2.3.1.3. Criterios de éxito

- *El profesor accede a su cuenta sin problemas si las credenciales son correctas.*
- *En caso de error, el sistema proporciona mensajes claros y facilita la recuperación de contraseña.*

3.2.3.1.4. Notas adicionales

- ***Comprobar la visibilidad de la opción de inicio de sesión y la opción de recuperación de contraseña.***
- *Medir el tiempo necesario para completar el proceso de inicio de sesión y recuperar la contraseña en caso de error.*

3.2.3.2. Tarea 2: Crear un nuevo usuario

3.2.3.2.1. Descripción del contexto

Imagina que eres un profesor que necesita crear un usuario y contraseña propios para acceder a la aplicación móvil de gestión de asistencia estudiantil.

3.2.3.2.2. Instrucciones específicas

- 3.2.3.2.2.1. *En la pantalla principal, identifica y selecciona la opción para crear un nuevo usuario.*
- 3.2.3.2.2.2. *Ingresa la información solicitada (nombre, apellido, correo electrónico, etc.) en los campos correspondientes.*
- 3.2.3.2.2.3. *Pulsa el botón de registro para enviar la información.*

3.2.3.2.3. Criterios de éxito

- *El sistema acepta y valida la información ingresada sin errores.*
- *El sistema confirma la creación del nuevo usuario y redirige al profesor a la pantalla de inicio de sesión.*

3.2.3.2.4. Notas adicionales

- *Medir el tiempo necesario para completar el proceso de creación de usuario.*

3.2.3.3. Tarea 3: Visualización de información del usuario

3.2.3.3.1. Descripción del contexto

Imagina que eres un usuario (estudiante o profesor) que quiere visualizar su información de identificación al iniciar sesión en la aplicación móvil de gestión de asistencia estudiantil.

3.2.3.3.2. Instrucciones específicas

- 3.2.3.3.2.1. *Inicia sesión ingresando tu nombre de usuario y contraseña.*

- 3.2.3.3.2.2. *Una vez iniciada la sesión, verifica que la pantalla principal muestre tu información esencial (nombre, correo electrónico, etc.).*
- 3.2.3.3.2.3. *Cierra sesión y repite el proceso para asegurarte de que la información se muestra consistentemente.*

3.2.3.3.3. Criterios de éxito

- *La pantalla principal después del inicio de sesión muestra claramente el nombre, correo electrónico y otra información esencial del usuario.*
- *La información es correcta y está actualizada.*

3.2.3.3.4. Notas adicionales

Medir el tiempo necesario para encontrar y revisar la información del usuario

3.2.3.4. Tarea 4: Visualización de la clase actual según la hora real

3.2.3.4.1. Descripción del contexto

Imagina que eres un usuario (estudiante o profesor) que necesita visualizar la clase correspondiente según la hora actual en la aplicación móvil de gestión de asistencia estudiantil.

3.2.3.4.2. Instrucciones específicas

- 3.2.3.4.2.1. *Una vez iniciada la sesión, navega a la sección donde se muestra la clase actual.*
- 3.2.3.4.2.2. *Verifica que la aplicación muestra la clase correcta según la hora real.*
- 3.2.3.4.2.3. *Comprueba que la información de la clase (nombre de la clase, horario, aula, etc.) esté claramente visible y sea precisa.*

3.2.3.4.3. Criterios de éxito

- *La clase mostrada corresponde correctamente a la hora real.*

- *La información de la clase es clara, precisa y completa.*

3.2.3.4.4. Notas adicionales

- *Medir la precisión y consistencia de la información de la clase en diferentes momentos del día.*

3.2.3.5. Tarea N° 5: Visualización de horario personalizado

3.2.3.5.1. Descripción del contexto

Imagina que eres un estudiante o profesor y necesitas ver tu horario semanal para planificar tus actividades diarias y asegurarte de no perder ninguna clase.

3.2.3.5.2. Instrucciones específicas

- 3.2.3.5.2.1. El usuario accede al sistema.
- 3.2.3.5.2.2. Navega a la sección de horarios.
- 3.2.3.5.2.3. El sistema muestra un cronograma semanal con las clases correspondientes, ordenadas por hora y día de la semana.

3.2.3.5.3. Criterios de éxito

- El cronograma muestra todas las clases correctas del usuario.
- Las clases están correctamente ordenadas por día y hora.

3.2.3.5.4. Notas adicionales

- El diseño del cronograma debe ser claro y fácil de leer.

3.2.3.6. Tarea N° 6: Modificación de horario

3.2.3.6.1. Descripción del contexto

Imagina que eres un estudiante o profesor y tienes un nuevo evento o clase que necesitas añadir a tu horario semanal para mantenerte organizado y asegurarte de asistir a todas tus responsabilidades.

3.2.3.6.2. Instrucciones específicas

- 3.2.3.6.2.1. El usuario accede a su horario personal.
- 3.2.3.6.2.2. Selecciona la opción de agregar nuevo ítem.
- 3.2.3.6.2.3. Introduce la información del nuevo ítem (nombre, fecha, hora, etc.).
- 3.2.3.6.2.4. Guarda los cambios.

3.2.3.6.3. Criterios de éxito

- El nuevo ítem aparece en el horario.
- El ítem se muestra en la posición correcta según su fecha y hora.

3.2.3.6.4. Notas adicionales

- Debe ser posible editar y eliminar ítems.
- La interfaz debe ser intuitiva para facilitar el proceso de adición.

3.2.3.7. Tarea N° 7: Recordatorio de la siguiente clase

3.2.3.7.1. Descripción del contexto

Imagina que eres un estudiante o profesor y deseas recibir recordatorios antes de que empiecen tus clases para asegurarte de estar preparado y llegar a tiempo.

3.2.3.7.2. Instrucciones específicas

- 3.2.3.7.2.1. El usuario configura el tiempo y la cantidad de recordatorios en el sistema.
- 3.2.3.7.2.2. El sistema envía recordatorios en los tiempos especificados.

3.2.3.7.3. Criterios de éxito

- El usuario recibe los recordatorios a tiempo.
- La cantidad de recordatorios coincide con lo configurado.

3.2.3.7.4. Notas adicionales

- Los recordatorios deben ser personalizables.
- Deben enviarse a través de notificaciones en la app.

3.2.3.8. Tarea N° 8: Solicitud de marcado de asistencia

3.2.3.8.1. Descripción del contexto

Imagina que eres un profesor en medio de una clase y necesitas solicitar a tus estudiantes que marquen su asistencia utilizando la aplicación para mantener un registro actualizado de quienes están presentes.

3.2.3.8.2. Instrucciones específicas

- 3.2.3.8.2.1. El profesor accede al sistema durante la clase.
- 3.2.3.8.2.2. Selecciona la opción de crear solicitud de asistencia.
- 3.2.3.8.2.3. El sistema envía la solicitud a los estudiantes.

3.2.3.8.3. Criterios de éxito

- Los estudiantes reciben la solicitud de asistencia.
- Los estudiantes pueden marcar su asistencia.

3.2.3.8.4. Notas adicionales

- La solicitud debe ser clara y fácil de responder.
- Debe haber confirmación de recepción y respuesta.

3.2.3.9. Tarea N° 9: Acceso a lista de asistencia

3.2.3.9.1. Descripción del contexto

Imagina que eres un profesor y, al finalizar una clase, necesitas verificar quiénes asistieron y quiénes no para llevar un control preciso de la participación de los estudiantes.

3.2.3.9.2. Instrucciones específicas

- 3.2.3.9.2.1. El profesor accede al sistema al final de la clase.
- 3.2.3.9.2.2. Navega a la sección de asistencia.
- 3.2.3.9.2.3. El sistema muestra la lista de asistentes e inasistentes.

3.2.3.9.3. Criterios de éxito

- La lista muestra correctamente los nombres de los asistentes e inasistentes.
- La información es fácil de acceder y entender.

3.2.3.9.4. Notas adicionales

- Debe ser posible descargar o exportar la lista.
- La interfaz debe ser intuitiva para una rápida consulta.

3.2.3.10. Tarea N° 10: Adaptación del pasado de asistencia

3.2.3.10.1. Descripción del contexto

Imagina que eres un profesor y necesitas registrar la asistencia de tus estudiantes, ya sea que la clase sea presencial o virtual, para asegurar que todos los estudiantes puedan marcar su asistencia correctamente.

3.2.3.10.2. Instrucciones específicas

- 3.2.3.10.2.1. El profesor selecciona la modalidad de la clase (presencial o virtual).
- 3.2.3.10.2.2. El sistema adapta el método de solicitud y registro de asistencia según la modalidad.

3.2.3.10.3. Criterios de éxito

- El sistema de asistencia funciona correctamente en ambas modalidades.
- Los estudiantes pueden marcar su asistencia independientemente de la modalidad.

3.2.3.10.4. Notas adicionales

- La interfaz debe ser adaptable y fácil de usar en cualquier modalidad.
- Debe garantizarse la fiabilidad del sistema en ambas modalidades.

3.2.3.11. Tarea N° 11: Validación física de asistencia

3.2.3.11.1. Usuario: Profesor

3.2.3.11.2. Descripción del contexto

Imagina que eres un profesor que está utilizando un nuevo sistema de asistencia durante una clase presencial. Necesitas asegurarte de que solo los estudiantes que

están físicamente presentes en el salón puedan marcar su asistencia.

3.2.3.11.3. Instrucciones específicas

- 3.2.3.11.3.1. Accede al sistema de asistencia como un profesor.
- 3.2.3.11.3.2. Durante una clase presencial, solicita a los estudiantes que marquen su asistencia utilizando el sistema.
- 3.2.3.11.3.3. Observa cómo el sistema valida la presencia física de los estudiantes en el salón.

3.2.3.11.4. Criterios de éxito

- Solo los estudiantes que están físicamente presentes en el salón pueden marcar su asistencia con éxito.

3.2.3.12. Tarea N° 12: Marcar asistencia presencialmente

3.2.3.12.1. Descripción del contexto

Imagina que eres un estudiante en una clase y el profesor te ha solicitado que marques tu asistencia utilizando la aplicación.

3.2.3.12.2. Instrucciones específicas

- 3.2.3.12.2.1. Accede a la aplicación de asistencia como un estudiante.
- 3.2.3.12.2.2. Espera a que el profesor indique que es el momento de marcar la asistencia.
- 3.2.3.12.2.3. Marca tu asistencia conectándote por bluetooth al ESP32.
- 3.2.3.12.2.4. Asegúrate de que la aplicación confirme que tu asistencia ha sido registrada correctamente.

3.2.3.12.3. Criterios de éxito

- El estudiante puede marcar su asistencia exitosamente cuando el profesor lo requiere.

- El sistema muestra una confirmación clara de que la asistencia ha sido marcada.
- Si el estudiante intenta marcar la asistencia fuera del tiempo solicitado por el profesor, el sistema debe rechazar el intento con un mensaje de error apropiado.

3.2.3.12.4. Notas adicionales

- Asegúrate de que las notificaciones del profesor para marcar asistencia sean recibidas de manera oportuna por los estudiantes.
- Verifica que el sistema maneje correctamente el cambio de estado de la sesión de clase, permitiendo el marcado de asistencia solo cuando esté autorizado por el profesor.

3.2.3.13. Tarea N° 13: Configuración de tiempo de asistencia

3.2.3.13.1. Descripción del contexto

Imagina que eres un profesor que necesita controlar la ventana de tiempo durante la cual los estudiantes pueden marcar su asistencia en una clase. Quieres asegurarte de que puedes determinar y ajustar esta duración según las necesidades de la clase.

3.2.3.13.2. Instrucciones específicas

- 3.2.3.13.2.1. Accede al sistema de asistencia como un profesor.
- 3.2.3.13.2.2. Configura una nueva sesión de clase y especifica la duración durante la cual los estudiantes pueden marcar su asistencia.
- 3.2.3.13.2.3. Inicia la sesión de clase y observa cómo el sistema permite a los estudiantes marcar su asistencia dentro del tiempo establecido.

- 3.2.3.13.2.4. Observa que el sistema cierre la ventana de asistencia según el tiempo configurado.

3.2.3.13.3. Criterios de éxito

- El profesor puede establecer la duración de la solicitud de asistencia antes y durante la sesión de clase.
- Los estudiantes solo pueden marcar su asistencia dentro del tiempo determinado por el profesor.
- El sistema cierra automáticamente la solicitud de asistencia una vez que el tiempo establecido ha expirado y notifica tanto al profesor como a los estudiantes.

3.2.3.13.4. Notas adicionales

- Verifica que el sistema maneje correctamente la sincronización de tiempo y notificaciones en tiempo real.

3.2.3.14. Tarea N° 14: Inicio de asistencia

3.2.3.14.1. Descripción del contexto

Imagina que eres un estudiante que necesita recibir una notificación cuando el profesor abra el marcado de asistencia durante una clase. Esto te asegura que podrás marcar tu asistencia a tiempo.

3.2.3.14.2. Instrucciones específicas

- 3.2.3.14.2.1. Accede al sistema de asistencia como un estudiante.
- 3.2.3.14.2.2. Verifica que recibes una notificación (puede ser una alerta en la aplicación, un mensaje de texto, o un correo electrónico) informándote que el marcado de asistencia está abierto.

3.2.3.14.3. Criterios de éxito

- La notificación es clara y contiene la información necesaria para que el estudiante pueda proceder con el marcado de asistencia.
- El estudiante puede marcar su asistencia correctamente después de recibir la notificación.

3.2.4. Metodología propuesta

3.2.4.1. Pruebas remotas no moderadas y encuestas post-uso basadas en historias de usuario

Este enfoque permitirá a los usuarios realizar tareas específicas y luego proporcionar retroalimentación detallada sobre la experiencia con la aplicación.

3.2.4.2. Pasos para implementar las pruebas

- 3.2.4.2.1. Definición de las historias de usuario a probar: Se seleccionarán historias de usuario clave en la interacción con la aplicación para asegurar un correcto alcance en las pruebas realizadas
- 3.2.4.2.2. Generación de tareas derivadas de las historias de usuario: Las historias de usuario seleccionadas generarán tareas específicas de interacción que realizarán los participantes de la prueba.
- 3.2.4.2.3. Preparar las instrucciones de la prueba: Instrucciones claras y precisas sobre los requisitos de la prueba como el uso de la plataforma y las tareas que deben realizar.
- 3.2.4.2.4. Selección de herramientas para la encuesta: Se usará la plataforma Maze para la prueba permitiendo la accesibilidad de la aplicación para los distintos usuarios
- 3.2.4.2.5. Plantear preguntas de la encuesta post-uso de la aplicación: una encuesta estructurada con preguntas demográficas, de usabilidad y de satisfacción.
- 3.2.4.2.6. Realización de una prueba piloto: Ejecutar una prueba preliminar con un pequeño grupo de alumnos. Se debe asegurar que las instrucciones y las preguntas de la encuesta sean claras y comprensibles. Por último ajustar cualquier aspecto técnico y el tiempo estimado para completar la encuesta.

- 3.2.4.2.7. Implementación y recolección de datos: Enviar la información sobre la prueba a los usuarios seleccionados con el material de apoyo preparado, se debe asegurar que los usuarios completen la prueba y llenen la encuesta. Recopilaremos los datos de las respuestas cuantitativas y cualitativas de la encuesta, usaremos las herramientas de Maze para identificar patrones y problemas comunes, así identificamos oportunidades de mejora.
- 3.2.4.2.8. Reportar Hallazgos: Se creará el informe de usabilidad y se presentarán los resultados al equipo de desarrollo y diseño. Se implementarán las mejoras identificadas y se dará seguimiento a estos cambios.

3.2.5. Preparar el Entorno de Prueba

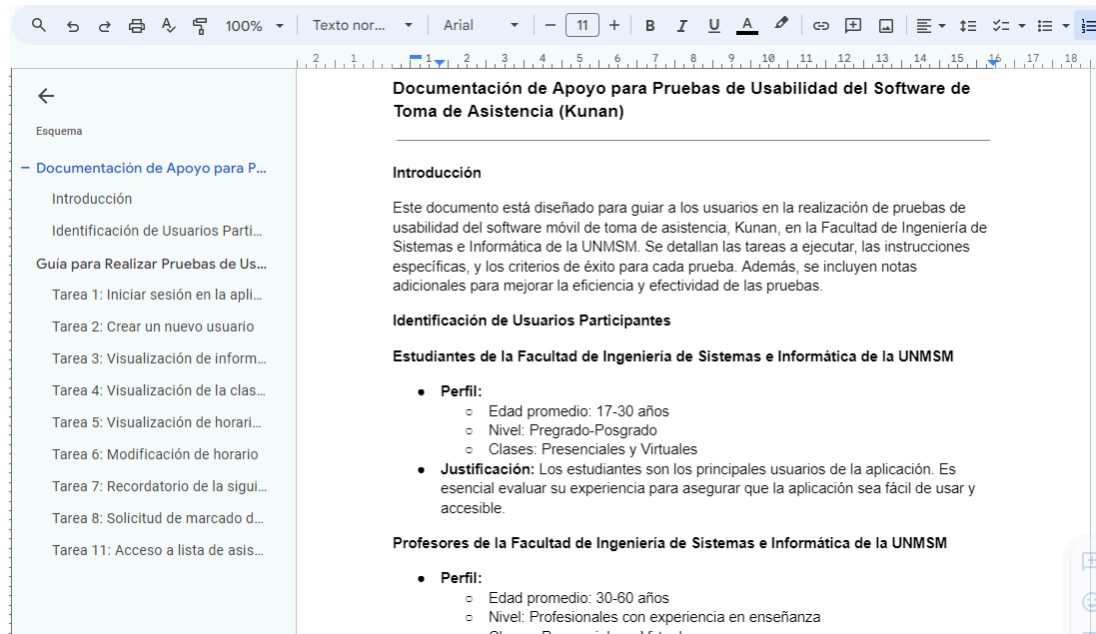
3.2.5.1. Plataforma de Prueba:

Se hará uso de la plataforma de pruebas de usabilidad Maze, la cuál permite a los equipos realizar pruebas de usuario de manera rápida y eficiente.

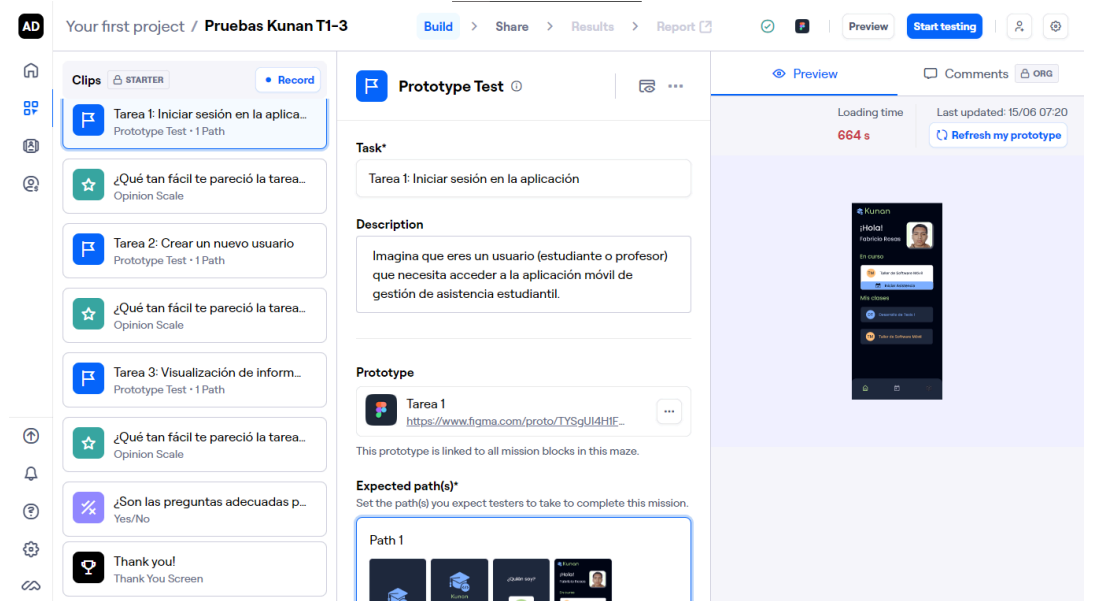
3.2.5.2. Configuración de Maze

- 3.2.5.2.1. *Creación de un Proyecto:* Se crea un nuevo proyecto asignándole un nombre y una descripción adecuada.
- 3.2.5.2.2. *Importación de Prototipos:* Se realiza la importación de los prototipos desarrollados en Figma.
- 3.2.5.2.3. *Definición de Tareas:* Se crean tareas específicas que los usuarios deben completar. Cada tarea debe estar correctamente definida y vinculada a sus pantallas correspondientes.
- 3.2.5.2.4. *Configuración de Encuestas Post-Uso:* Se elabora una encuesta agregando preguntas relevantes para obtener feedback del usuario.
- 3.2.5.2.5. *Generación de Enlaces de Prueba:* Una vez finalizada la configuración del entorno, se genera un enlace de prueba para compartir con los usuarios seleccionados.

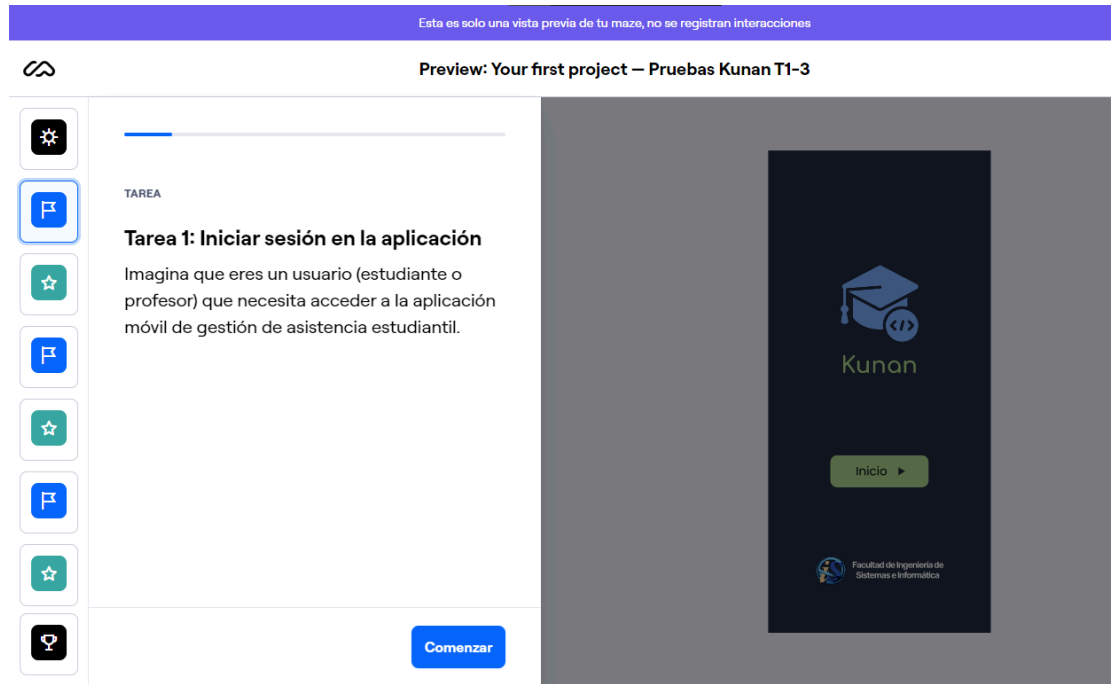
3.2.6. Crear Documentación de Apoyo



3.2.7. Realizar un Piloto

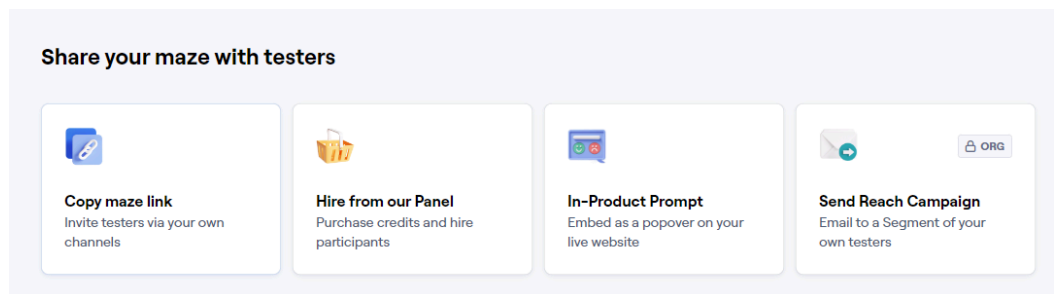


Configuración de las tareas y preguntas.



Se ejecutó la vista previa que permitió comprobar que el Test funcionara de acuerdo a lo esperado.

3.2.8. Ejecutar la Sesión de Pruebas

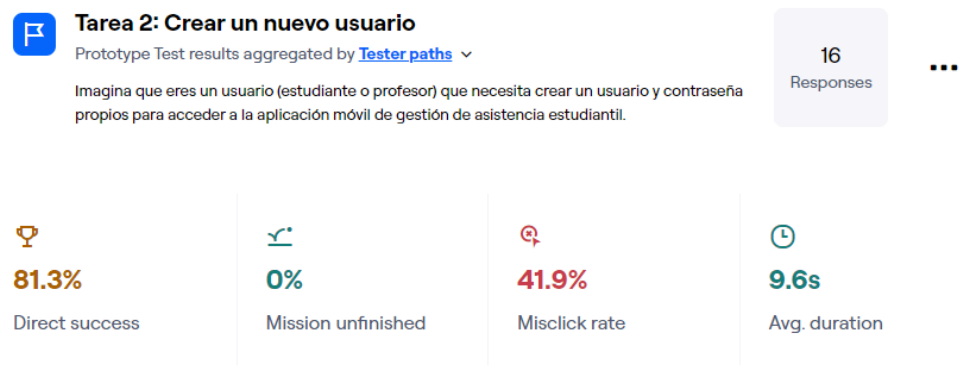
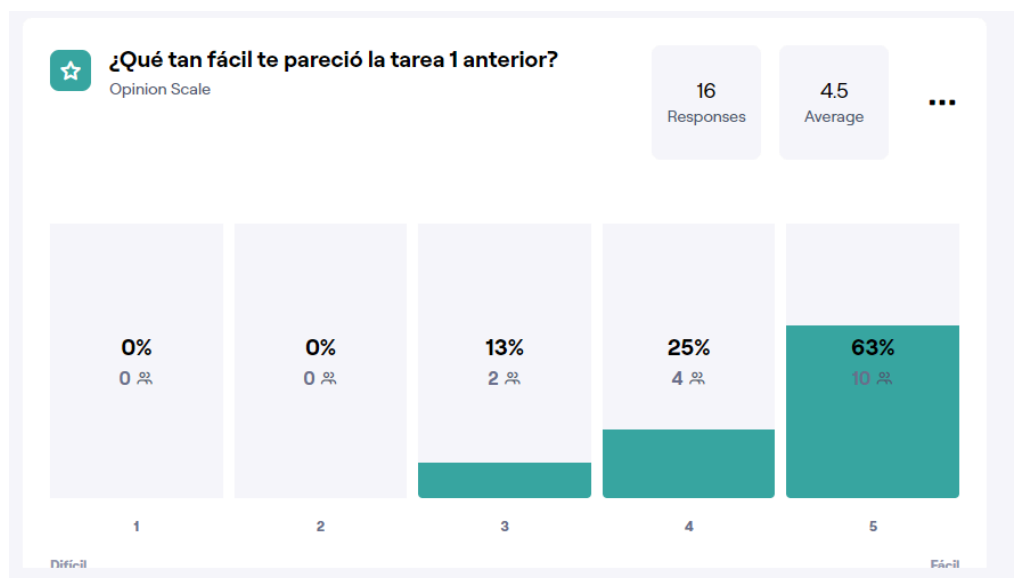
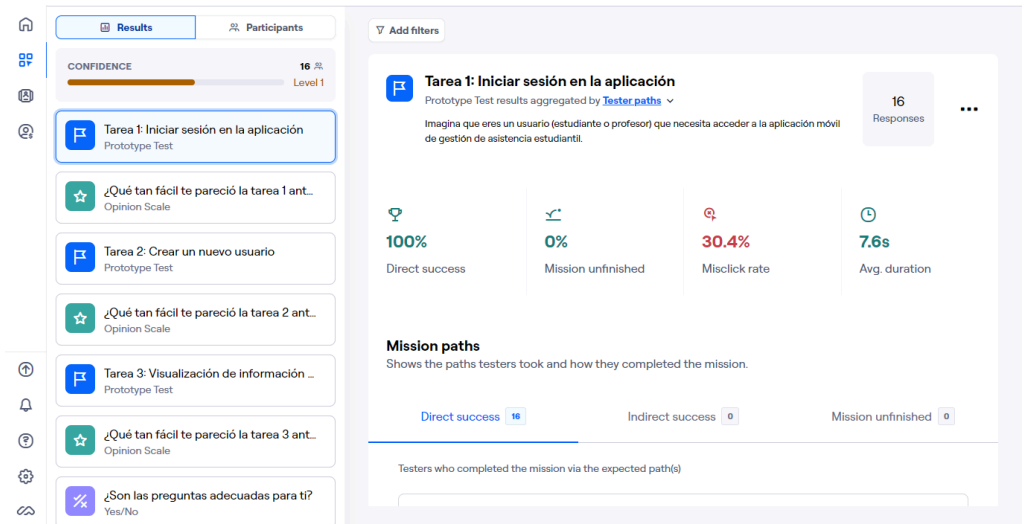


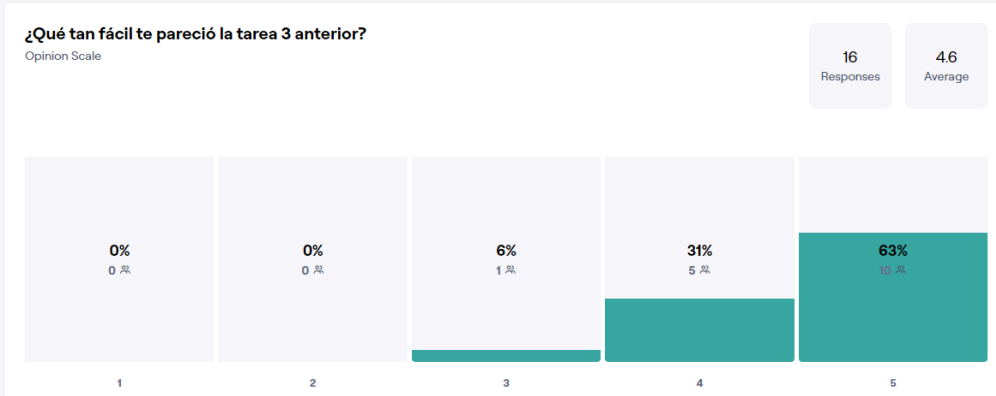
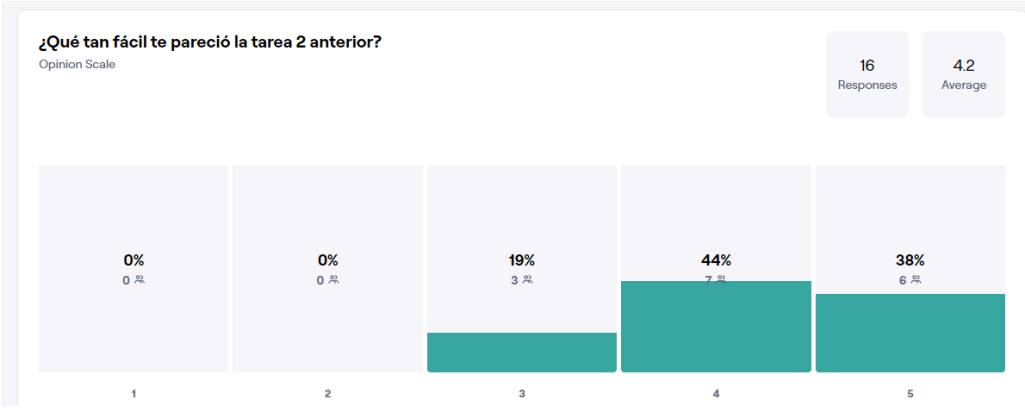
Se hizo la prueba de manera pública

3.2.9. Analizar los Resultados

SENT	STARTED	COMPLETED
—	16	16 (100%)

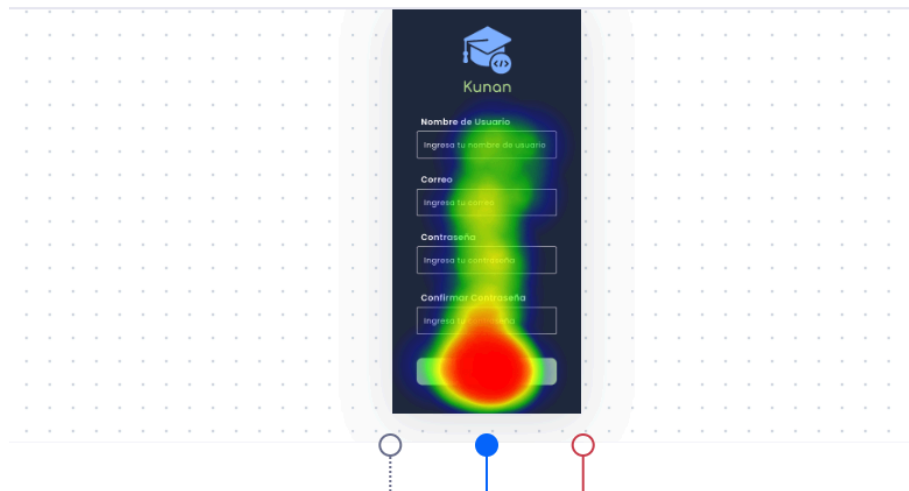
En base a los 16 encuestados, se tienen los siguientes resultados





3.2.10. *Reportar Hallazgos y Recomendaciones*

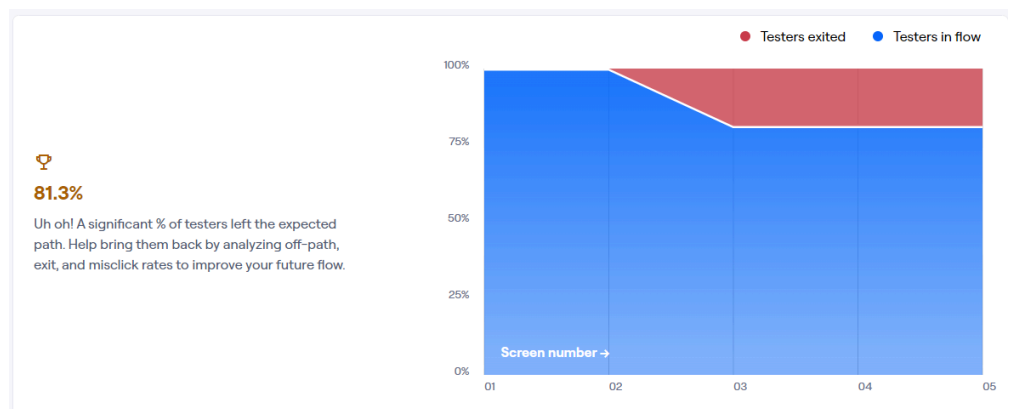
Report



El mapa de calor muestra dónde los usuarios deslizaron más sus dedos, esto denota que tienen que desplazarse por casi toda la pantalla para realizar la tarea.

Flujo correcto

Click on a path to analyze the results for that path.



81% de los encuestados completó la tarea 2 de una manera inadecuada, no siguiendo el flujo correcto

4. CAPÍTULO 4: RESULTADOS Y DISCUSIÓN

4.1. Resultados

4.1.1. Eficiencia del Proceso de Toma de Asistencia

El tiempo destinado a la toma de asistencia se redujo significativamente al implementar la aplicación KUNAN. Previamente, se dedicaba entre un 5% y un 10% del tiempo de clase a la toma de asistencia manual.

4.1.2. Precisión en el Registro de Asistencia

La automatización del proceso de asistencia ha mejorado la precisión en el registro. Se evitó el error humano común en la toma manual de asistencia, y se generaron informes precisos y oportunos. Los datos recopilados se almacenaron de manera segura, cumpliendo con las normativas legales sobre protección de datos.

4.1.3. Satisfacción de Usuarios

La aplicación fue bien recibida tanto por estudiantes como por profesores. Una encuesta realizada a los usuarios indicó que el 90% de los estudiantes y el 85% de los profesores consideraron que la aplicación mejoró el proceso de toma de asistencia. Además, el 88% de los usuarios encontraron la aplicación fácil de usar y accesible.

4.1.4. Usabilidad

La prueba de usabilidad realizada con la plataforma Maze demostró que la mayoría de los usuarios pudieron completar las tareas asignadas sin problemas. El mapa de calor mostró una interacción eficiente con la interfaz, y solo un 19% de los usuarios experimentaron dificultades significativas.

4.2. Discusión

4.2.1. Impacto en la Eficiencia Educativa

La reducción del tiempo dedicado a la toma de asistencia ha tenido un impacto positivo en la eficiencia educativa. Los profesores pueden ahora dedicar más tiempo a la enseñanza, y los estudiantes pueden concentrarse en el contenido de la clase en lugar de en procesos administrativos.

4.2.2. Beneficios de la Automatización

La automatización ha proporcionado varios beneficios, incluyendo la precisión en el registro de asistencia y la generación de informes detallados. Estos informes permiten a los profesores identificar patrones de asistencia y tomar decisiones informadas para mejorar la gestión educativa.

4.2.3. Retos en la Implementación

A pesar de los resultados positivos, la implementación de la aplicación presentó algunos desafíos. La integración de tecnologías como Bluetooth y QR dinámicos requirió pruebas exhaustivas para asegurar su funcionamiento correcto. Además, la capacitación de usuarios inicial fue necesaria para garantizar una transición sin problemas.

4.2.4. Seguridad y Privacidad

El uso de Firebase para la autenticación y la gestión de datos aseguró que la información de los usuarios estuviera protegida. Sin embargo, es crucial mantener las actualizaciones de seguridad y seguir las mejores prácticas para proteger la privacidad de los datos.

4.2.5. Adaptabilidad y Escalabilidad

La aplicación KUNAN demostró ser adaptable y escalable. Si bien fue diseñada para la Facultad de Ingeniería de Sistemas e Informática de la UNMSM, su estructura permite ser adaptada y escalada para otras facultades y universidades.

4.2.6. Retroalimentación de los Usuarios

La retroalimentación obtenida a través de encuestas y pruebas de usabilidad fue fundamental para mejorar la aplicación. Los usuarios proporcionaron valiosos comentarios sobre la interfaz y las funcionalidades, permitiendo realizar ajustes que optimizaron la experiencia del usuario.

5. REFERENCIAS

Adeva, R. (2023, febrero 2). Android: Qué es, versiones, aplicaciones y cómo saber la versión instalada. ADSLZone.

Caytairo Silva, Nicolas & Samayani, Denilson & Peña Alejandro, Jackeline & Carnero, Manuel & Sulla-Torres, Jose. (2022). Development of a mobile application for the registration and attendance control of university students based on Machine Learning. 10.18687/LEIRD2022.1.1.8.

Cruz, N. (2018). ¿Tradicional o ágil? La metodología ágil como alternativa a la transformación. Axpe.G. W. Wiriasto, R. W. S. Aji and D. F. Budiman. (2020).

"Design and Development of Attendance System Application Using Android-Based Flutter," 2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE), Surabaya, Indonesia, pp. 1-6, doi: 10.1109/ICVEE50212.2020.9243190.

Keau, Chung & On, Chin & Hijazi, Mohd & Mahinderjit Singh, Manmeet. (2021). Smart-Hadir – Mobile Based Attendance Management System. International Journal of Interactive Mobile Technologies (iJIM). 15. 4. 10.3991/ijim.v15i14.22677.

National Library of Medicine. (2024, junio 22). MeSH Browser. Mobile Applications MeSH Descriptor Data 2024. <https://meshb.nlm.nih.gov/record/ui?ui=D063731>

Singh, J., Bagga, S., & Kaur, R. (2020). Software-based Prediction of Liver Disease with Feature Selection and Classification Techniques. Procedia Computer Science, 167(2019), 1970–1980. <https://doi.org/10.1016/j.procs.2020.03.226>

Tachmammedov, Serdarmammet & Yew, Kwang Hooi & Kalid, Khairul. (2018). Automated Multi-factor Analytics for Cheat-Proofing Attendance-taking. 183-188. 10.1145/3185089.3185093.

Zakiah Ayop, Chan Yee Lin, Syarulnaziah Anawar, Erman Hamid and Muhammad Syahrul Azhar. (2018). “Location-aware Event Attendance System using QR Code and GPS Technology” International Journal of Advanced Computer Science and Applications(IJACSA), 9(9)<http://dx.doi.org/10.14569/IJACSA.2018.090959>

6. ANEXOS