



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Tema:

APLICACIÓN MÓVIL USANDO EL FRAMEWORK FLUTTER PARA EL
CONTROL DE LAS TUTORÍAS ACADÉMICAS DEL SISTEMA
INTEGRADO DE LA UNIVERSIDAD TÉCNICA DE AMBATO

Trabajo de Integración Curricular Modalidad: Proyecto de Investigación, presentado previo a la obtención del Título de Ingeniera en Tecnologías de la Información.

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de software

AUTOR: Paola Estefanía Bautista Salazar

TUTOR: Ing. Clay Fernando Aldás Flores, Mg.

Ambato – Ecuador

marzo - 2022

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Integración Curricular con el tema: APLICACIÓN MÓVIL USANDO EL FRAMEWORK FLUTTER PARA EL CONTROL DE LAS TUTORÍAS ACADÉMICAS DEL SISTEMA INTEGRADO DE LA UNIVERSIDAD TÉCNICA DE AMBATO, desarrollado bajo la modalidad de Proyecto de Investigación por la señorita Paola Estefanía Bautista Salazar, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo el trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la ejecución de la Unidad de Integración Curricular y la obtención del título de tercer nivel, de grado en la Universidad Técnica de Ambato y sus reformas y el numeral 7.4 del respectivo instructivo.

Ambato, marzo 2022

TUTOR

AUTORÍA

El presente trabajo de Integración Curricular titulado: APLICACIÓN MÓVIL USANDO EL FRAMEWORK FLUTTER PARA EL CONTROL DE LAS TUTORÍAS ACADÉMICAS DEL SISTEMA INTEGRADO DE LA UNIVERSIDAD TÉCNICA DE AMBATO, es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, marzo 2022.



Paola Estefanía Bautista Salazar

C. C. 1805177027

AUTORA

APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Integración Curricular presentado por la señorita Paola Estefanía Bautista Salazar, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad de Proyecto de Investigación, titulado **APLICACIÓN MÓVIL USANDO EL FRAMEWORK FLUTTER PARA EL CONTROL DE LAS TUTORÍAS ACADÉMICAS DEL SISTEMA INTEGRADO DE LA UNIVERSIDAD TÉCNICA DE AMBATO**, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la ejecución de la Unidad de Integración Curricular y la obtención del título de tercer nivel, de grado en la Universidad Técnica de Ambato y sus reformas y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, marzo 2022.

Ing. Pilar Urrutia, Mg.

PRESIDENTE DEL TRIBUNAL

Ing. Carlos Núñez

PROFESOR CALIFICADOR

Ing. Leonardo Torres

PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Integración Curricular como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Integración Curricular en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, marzo 2022.



Paola Estefanía Bautista Salazar

C. C. 1805177027

AUTORA

DEDICATORIA

El presente proyecto es dedicado a mis padres, María y Gerardo por ser las personas más importantes en mi vida, por su constante apoyo, dedicación, amor, paciencia y por estar en mis logros y alentarme a no rendirme ante mis fracasos. Gracias por ser los mejores padres.

A mi hermano Víctor por ser el mejor profesor que me ha enseñado demasiado y ser mi modelo a seguir, por instruirme, acompañarme y estudiar conmigo durante las noches de desvelo en el proceso de mi formación profesional. Gracias por creer en mí.

A mi hermana Belén, por cuidarme cuando era niña, por ser mi mejor amiga, confidente y la persona que siempre me escucha y porque ha estado conmigo en las buenas y en las malas. Por ti he aprendido muchísimo para ser una mejor persona.

A mis sobrinas Doménica y Bia por ser las personas más dulces que alegran mi vida con solo sonreír y con las que disfruto compartir tiempo.

A Tess y Chiri por ser mis compañeras peluditas de vida.

Paola Estefanía Bautista Salazar

AGRADECIMIENTO

Agradezco a la Dirección Académica (DAC) y a la Dirección de Tecnologías de Información y Comunicación (DITIC) de la Universidad Técnica de Ambato por haberme abierto las puertas para realizar el presente proyecto.

A los docentes de la facultad por su entrega, esfuerzo y compromiso como profesionales. Gracias por haberme brindado sus valiosos conocimientos y alentarme para ser una buena profesional.

Un agradecimiento especial a mi tutor Ing. Clay Aldás no solo por guiarme durante el desarrollo de este proyecto sino por preocuparse, compartir e impartir sus conocimientos dentro y fuera del aula. Por ser un ser excepcional, gracias por todo.

ÍNDICE DE CONTENIDOS

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
APROBACIÓN TRIBUNAL DE GRADO.....	iv
DERECHOS DE AUTOR	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
RESUMEN EJECUTIVO	xvii
ABSTRACT	xviii
CAPÍTULO 1	1
MARCO TEÓRICO.....	1
1.1. Tema.....	1
1.2. Antecedentes investigativos	1
1.2.1. Contextualización.....	1
1.2.2. Fundamentación teórica	2
1.3. Objetivos	8
1.3.1. Objetivo general	8
1.3.2. Objetivos específicos	8
CAPÍTULO II	9
METODOLOGÍA	9
2.1. Materiales.....	9

2.2.	Métodos.....	17
2.2.1.	Modalidad de investigación	17
2.2.2.	Población y muestra	17
2.2.3.	Recolección de información.....	19
CAPÍTULO III.....		38
RESULTADOS Y DISCUSIÓN		38
3.1.	Análisis y discusión de resultados.....	38
3.1.1.	Flutter	38
3.1.2.	Arquitectura de la aplicación móvil	40
3.1.3.	Metodología de desarrollo.....	40
3.2.	Desarrollo de la propuesta.....	43
3.2.1.	Fase I: Exploración	43
3.2.2.	Fase II: Inicialización.....	45
3.2.3.	Fases III: Producción	64
3.2.4.	Fase IV: Estabilización	91
3.2.5.	Fase V: Pruebas.....	91
CAPÍTULO IV.....		106
CONCLUSIONES Y RECOMENDACIONES.....		106
4.1.	Conclusiones	106
4.2.	Recomendaciones.....	107
BIBLIOGRAFÍA		108

ANEXOS	110
--------------	-----

ÍNDICE DE FIGURAS

Figura 1: Frecuencia de solicitudes en horarios asignados para las Tutorías Académicas	21
Figura 2: Periodicidad de asistencia de Tutorías Académicas a los estudiantes.	22
Figura 3: Frecuencia de que el docente le recuerda a los estudiantes de validar las Tutorías Académicas	23
Figura 4: Aprobación de los docentes sobre una App Móvil para mejorar la interactividad de las Tutorías Académicas.....	24
Figura 5: Aprobación de docentes encuestados sobre una app móvil para optimizar el registro de las Tutorías Académicas.....	25
Figura 6: Importancia de que se le notifique al estudiante que debe validar tutorías luego de haberlas asistido	26
Figura 7: Aceptación de tutorar a los estudiantes en horas de actividades académicas.	27
Figura 8: Notificaciones a los estudiantes sobre las Tutorías Académicas por medio de la App Móvil	28
Figura 9: Solicitud de Tutorías Académicas en horas asignadas.....	29
Figura 10: Frecuencia de solicitudes para Tutorías Académicas en horas asignadas	30
Figura 11: Frecuencia de solicitudes para Tutorías Académicas fuera del horario asignado	31
Figura 12: Aprobación de estudiantes encuestados para mejorar la interactividad de las Tutorías Académicas	32
Figura 13: Conformidad de estudiantes encuestados para que la App Móvil pueda solicitar tutorías en cuando el docente tenga disponibilidad.....	33

Figura 14: Estudiantes encuestados que quieren notificaciones para validar luego de haber asistido a tutorías.....	34
Figura 15: Estudiantes encuestados que quieren notificaciones para validar a tiempo las tutorías antes de terminar cada parcial.....	35
Figura 16: Fases de la Metodología Mobile – D	42
Figura 17: Modelo Conceptual	49
Figura 18: Modelo Relacional	50
Figura 19: Arquitectura física de la Aplicación Móvil.....	51
Figura 20: Diagrama de caso de uso de los Estudiantes.....	58
Figura 21: Diagrama de caso de uso de los Docentes.....	58
Figura 22: Inicio de Sesión de usuarios y docentes.....	59
Figura 23: Menú de estudiantes	60
Figura 24: Solicitud de Tutorías	60
Figura 25: Registro de tutorías.....	61
Figura 26: Ver estados de tutorías	61
Figura 27: Menú para docentes.....	62
Figura 28: Registrar tutorías por parte de los docentes	63
Figura 29: Ver estados de las tutorías solicitadas al docente.....	63
Figura 30: Gestión de Tutorías Académicas.....	64
Figura 31: Servicio para actualizar una tutoría cuando un estudiante la valida.	68
Figura 32: Interfaz para el uso del método updateDetalleTutorias.....	68
Figura 33: Controlador para actualizar una tutoría cuando un estudiante la valida .	69
Figura 34: Comunicación entre frontend y backend para el inicio de Sesión	71
Figura 35: Código para obtener las materias donde el estudiante está matriculado .	72
Figura 36: Código para obtener horarios acordes a la materia	73

Figura 37: Código para insertar una solicitud de Tutoría Académica	74
Figura 38: Código de la consulta de las Tutorías del estudiante para validaciones..	75
Figura 39: Validación de tutorías.....	76
Figura 40: Código para ver el estado actual de las tutorías para el estudiante	77
Figura 41: Código para obtener las materias que el docente imparte.....	79
Figura 42: Código sobre la consulta de tutorías para docentes.....	80
Figura 43: Actualización de estados de tutorías para docentes	80
Figura 44: Listado de tutorías acorde a las validaciones de los estudiantes	82
Figura 45: Consumo de la API Firebase Cloud Messaging.....	83
Figura 46: Código - Notificación para el docente que tiene una nueva solicitud de tutoría	84
Figura 47: Código - Notificación para el estudiante que tiene una tutoría pendiente de validar.....	85
Figura 48: Código - Notificación en caso de que se rechace una solicitud de tutoría de un estudiante.....	86
Figura 49: Código - Notificación programada para que el estudiante valide una tutoría después de una hora.	88
Figura 50: Código - Notificación programada una semana antes de finalizar cada parcial.....	89
Figura 51: HostedService para tareas programadas.....	90
Figura 52: Inicio de sesión para docentes y estudiantes	92
Figura 53: Menú para estudiantes y docentes respectivamente.....	93
Figura 54: Solicitud de tutorías.....	94
Figura 55: Validación de Tutorías Académicas.....	95
Figura 56: Estados de las tutorías para estudiantes.....	96

Figura 57: Registro de Tutorías Académicas para los docentes	97
Figura 58: Gestión de Tutorías Académicas.....	98
Figura 59: Ver los estados de las Tutorías Académicas para docentes	99
Figura 60: Resultado de notificación cuando recibe una solicitud	99
Figura 61: Resultado de notificación una hora después de haber solicitado una tutoría	100
Figura 62: Notificación de solicitud rechazada	100
Figura 63: Resultado de notificación una semana antes de finalizar el parcial	101

ÍNDICE DE TABLAS

Tabla 1: Respuestas HTTP Informativas -----	5
Tabla 2: Respuestas HTTP exitosas -----	6
Tabla 3: Respuestas HTTP de redireccionamiento -----	6
Tabla 4: Respuestas HTTP Error del cliente -----	7
Tabla 5: Respuesta HTTP Error del servidor-----	7
Tabla 6: Población de docentes -----	17
Tabla 7: Población de estudiantes -----	18
Tabla 8: Características de Flutter -----	38
Tabla 9: Importancia del uso del Framework Flutter-----	39
Tabla 10: Metodologías Ágiles -----	41
Tabla 11: Planificación de fases -----	48
Tabla 12: Funcionalidades para los estudiantes-----	55
Tabla 13: Funcionalidades para el docente -----	57
Tabla 14: Storycard – Inicio de Sesión estudiantes.-----	70
Tabla 15: Storycard - Solicitar Tutorías Académicas -----	71
Tabla 16: Storycard - Registro de Tutorías Académicas-----	74
Tabla 17: Storycard - Ver estados de las tutorías solicitadas-----	76
Tabla 18: Storycard – Inicio de Sesión de docentes -----	77
Tabla 19: Storycard - Registro de Tutorías Académicas para docentes-----	78
Tabla 20: Storycard - Gestión de Estados de Tutorías Académicas-----	79
Tabla 21: Storycard - Ver estados de las Tutorías Académicas validadas-----	81
Tabla 22: Storycard - Notificaciones para docentes y estudiantes-----	82
Tabla 23: Prueba de aceptación 1 -----	101
Tabla 24: Prueba de aceptación 2 -----	102

Tabla 25: Prueba de aceptación 3-----	102
Tabla 26: Prueba de aceptación 4-----	103
Tabla 27: Prueba de aceptación 5-----	103
Tabla 28: Prueba de aceptación 6-----	104
Tabla 29: Prueba de aceptación 7-----	104
Tabla 30: Prueba de aceptación 8-----	105

RESUMEN EJECUTIVO

En la actualidad el uso de las aplicaciones móviles tiene un importante impacto en la sociedad. Existe una variedad de aplicaciones para todo tipo de interés y algunas de estas han servido como herramientas ya sea para estudios, trabajo u otras actividades del día a día.

El presente proyecto tiene el fin de ayudar con nuevas funcionalidades para optimizar las actividades que se realiza durante el proceso y mejorar la interacción entre los docentes y estudiantes con las Tutorías Académicas en la Universidad Técnica de Ambato.

La Aplicación Móvil fue creada en base a la Metodología Mobile - D consiguiendo flexibilidad y rapidez durante el desarrollo, utilizando el framework Flutter con el lenguaje de programación Dart, ASP.NET Core 5 como backend, que está conectado a la base de datos diseñada en SQL Server.

De esta manera, se logró cumplir el propósito principal de la Aplicación Móvil que es ser una herramienta útil para cumplir con las actividades académicas de los docentes y estudiantes, usando las Tecnologías de la Información por medio de los teléfonos inteligentes.

Palabras clave: Tecnologías de la información, aplicación móvil, tutorías académicas, metodología Mobile -D, framework Flutter.

ABSTRACT

Nowadays, the use of mobile applications has an important impact on society. There is a variety of applications for all types of interests and some of these have helped as tools either for studies, work or other day-to-day activities.

The present project aims to help with new functionalities to optimize the activities carried out during the process and improve the interaction between teachers and students with the Academic Tutorials at the Universidad Técnica de Ambato.

The Mobile Application was created based on the Mobile - D Methodology, achieving flexibility and swiftness during the use of the Flutter framework with the Dart programming language, ASP.NET Core 5 as a backend, that's connected to the database designed in SQL Server.

In this way, the main purpose of the Mobile Application was fulfilled, which is to be a useful tool to perform the academic activities of teachers and students using Information Technology through smartphones.

Key words: Information technology, mobile application, academic tutorials, Mobile-D methodology, Flutter framework.

CAPÍTULO 1

MARCO TEÓRICO

1.1. Tema

APLICACIÓN MÓVIL USANDO EL FRAMEWORK FLUTTER PARA EL CONTROL DE LAS TUTORÍAS ACADÉMICAS DEL SISTEMA INTEGRADO DE LA UNIVERSIDAD TÉCNICA DE AMBATO.

1.2. Antecedentes investigativos

1.2.1. Contextualización

Las Universidades Latinoamericanas a través de los años han ido buscando cómo mejorar sus servicios en el ámbito académico y administrativo conjunto con las Tecnologías de la Información puesto que han ido encontrando dificultades al momento de gestionar estos servicios ya sea de manera manual o en sistemas que han tenido suficiente carga de trabajo por tratar de implementar más funciones en el mismo, en consecuencia han acudido al desarrollo de los Sistemas Integrados para poder solucionar sus problemas. La Universidad para el Desarrollo Andino de Perú indica que buscó mejorar su Sistema Integrado con el fin de solucionar ciertos problemas en el control de las actividades académicas que fueron detectadas en su sistema actual ya que podría perjudicar a sus representados y docentes de dicha institución [1]. Por otra parte, la Universidad Tecnológica de Bolívar señala que han tenido problemas al mantener diversos sistemas de autenticación de identidad produciendo una complejidad en su manejo y el riesgo de seguridad con la confidencialidad de datos por lo que buscaron desarrollar un Sistema Integrado para corregir estos inconvenientes [2].

En algunas Universidades del Ecuador también hacen uso de Sistemas Integrados porque se vieron en la necesidad de mejorar sus servicios a causa de que no se manejaban de manera óptima. Un claro ejemplo de la necesidad de mejorar un Sistema Integrado, es porque se han detectado insuficiencias en el proceso de evaluación del docente universitario en la zona ocho de las Universidades del Ecuador [3]. También la Universidad Técnica del Norte de Ibarra encontró una escasa integración entre los

procesos bibliotecarios y que además limitaba la disponibilidad a los servicios causando insatisfacciones en los usuarios de su biblioteca por lo que buscaron desarrollar un módulo para mejorar la administración, flujo de información, servicios bibliotecarios, búsquedas, etc. [4]

En el caso de la Universidad Técnica de Ambato también cuenta con un Sistema Integrado que brinda servicios académicos y administrativos que de hecho otorgan muchos beneficios. Sin embargo se han encontrado ciertos inconvenientes con este módulo de tutorías académicas dado que se encuentra gestionado pero existen ciertas características que se podrían mejorar y en este caso tiene que ver con su uso restrictivo en vista de que no hay mucha interacción con este módulo, los estudiantes y docentes considerando que solo se puede validar en un tiempo limitado al final del parcial los registros de tutorías, más no solicitar o registrar una tutoría dependiendo de la fecha y tema que se esté llevando a cabo según conste en el sílabo. Además, los docentes deben registrar sus tutorías escribiendo los temas manualmente y no seleccionar por subunidades o temas tratados acorde al sílabo y las fechas en los que están contemplados.

Considerando los aspectos mencionados, estos han causado falta de interés por parte de los estudiantes con las tutorías y por lo tanto insatisfacción con el módulo por lo que se busca mejorarlo aplicando las Tecnologías de Información por medio de una aplicación móvil.

1.2.2. Fundamentación teórica

- **Ingeniería del Software**

Es una disciplina de la ingeniería que engloba los aspectos de producción de software desde las etapas iniciales de la especificación del sistema, su mantenimiento que comprende dos frases clave:

1. Disciplina de la ingeniería

Abarca todo lo que tiene que ver con las teorías, métodos y herramientas que sean convenientes para describir soluciones a problemas considerando las restricciones financieras y organizacionales.

2. Todos los aspectos de producción de software

La ingeniería de software no solo incluye los aspectos técnica de desarrollo de software, sino que también se consideran las actividades de gestión de proyectos de software y el desarrollo de herramientas, métodos y teorías de apoyo para la producción de software [5].

- **Aplicaciones móviles**

Son paquetes de software diseñadas para ejecutar en dispositivos móviles como smartphones, tablets, etc. Son distribuidas en plataformas por compañías propietaria de los sistemas operativos móviles como Play Store de Google para Android, App Store de Apple para iOS, entre otros. Existen aplicaciones gratuitas y pagadas.

Existen 3 tipos de aplicaciones:

- **Aplicaciones nativas**

Se crean para un sistema operativo determinado llamado Development Kit o SDK y se desarrolla con lenguajes de programación específicos por lo que las aplicaciones móviles nativas requieren más trabajo y tiempo.

Este tipo de aplicaciones aprovechan al máximo las características del dispositivo y las funcionalidades trabajan de manera óptima.

- **Web Apps**

Aplicaciones web desarrolladas en HTML, Javascript y CSS. Es independiente del sistema operativo y se puede ejecutar en diferentes dispositivos móviles sin crear varias aplicaciones.

- **Aplicaciones Híbridas**

Es la combinación de las aplicaciones anteriores. Permite el uso en diferentes plataformas. [6]

- **Framework Flutter**

Flutter es un kit de herramientas de desarrollo para crear aplicaciones móviles compiladas nativamente con un solo código desarrollado. Ofreciendo un gran número de librerías para las plataformas Android y iOS [7].

- **Web API**

Conjunto de interfaces definidas las cuales permiten a un conjunto de aplicaciones externas consumir el software de la aplicación web. Una API permite que dos aplicaciones completamente distintas puedan comunicarse sin ningún inconveniente.

- **REST (Representational State Transfer)**

Es una estructura que permite construir servicios web a los cuales se adhieren a un conjunto de principios establecidos [8].

- **Métodos HTTP**

Son mecanismos del protocolo HTTP que permiten expresar la acción de la cual se quiere ejercer sobre un recurso [8].

- **HTTP GET**

Sirve para consultar datos del servidor.

- **HTTP HEAD**

Hace lo mismo que el método GET, pero no retorna el cuerpo de la respuesta, sino solo la cabecera.

- **HTTP POST**

Sirve para solicitar la creación de un nuevo registro.

- **HTTP PUT**

Sirve para realizar actualizaciones de un registro.

- **HTTP DELETE**

Sirve para eliminar un registro.

- **HTTP PATCH**

Sirve para hacer actualizaciones parciales de un registro.

- **Códigos de estatus HTTP**

Son resultados que indican el estatus de respuesta de un servidor web [9].

- **1xx:** Respuestas informativas.

Código	Respuesta	Descripción
100	Continue	Respuesta que indica que hasta el momento está bien y que debe continuar con la solicitud o ignorarla.
101	Switching Protocol	Se envía a una respuesta a un encabezado de solicitud Upgrade (en-US) por el cliente e indica que el servidor acepta el cambio de protocolo propuesto por el agente de usuario.
102	Processing	Indica que el servidor recibió la solicitud y que se encuentra procesándola y que aún no hay una respuesta disponible.

Tabla 1: Respuestas HTTP Informativas

Elaborado por: Paola Bautista

- **2xx:** Respuestas exitosas [9].

Código	Respuesta	Descripción
200	Ok	La solicitud ha tenido éxito dependiendo del método HTTP.
201	Created	La solicitud ha tenido éxito y se ha creado un nuevo registro. Respuesta enviada por una petición PUT

202	Accepted	La solicitud fue enviada pero aún no se ha hecho nada.
204	No content	La petición se completó con éxito, pero su respuesta no tiene contenido.

Tabla 2: Respuestas HTTP exitosas

Elaborado por: Paola Bautista

- **3xx:** Respuestas de redireccionamiento [9].

Código	Respuesta	Descripción
301	Moved Permanently	Significa que la URI del recurso solicitado ha sido cambiado
302	Found	Significa que el recurso de la URI solicitada ha sido cambiado temporalmente.

Tabla 3: Respuestas HTTP de redireccionamiento

Elaborado por: Paola Bautista

- **4xx:** Error del cliente [9].

Código	Respuesta	Descripción
400	Continue	Significa que el servidor no pudo interpretar la solicitud dada una sintaxis inválida.
401	Unauthorized	Es necesario autenticarse para obtener una respuesta.

403	Forbidden	El cliente no posee los permisos necesarios para cierto contenido por lo que el servidor rechaza la petición.
404	Not Found	El servidor no pudo encontrar el contenido solicitado.
405	Method not allowed	El método solicitado ha sido deshabilitado pro el servidor.

Tabla 4: Respuestas HTTP Error del cliente

Elaborado por: Paola Bautista

- **5xx:** Error del servidor [9].

Código	Respuesta	Descripción
500	Internal Server Error	El servidor ha encontrado una situación que no sabe cómo manejarla.

Tabla 5: Respuesta HTTP Error del servidor

Elaborado por: Paola Bautista

- **Reglamento de Régimen Académico del Ecuador**

El reglamento regula y orienta el quehacer académico de las Instituciones de Educación Superior (IES) en sus diversos niveles de formación, incluyendo sus modalidades de aprendizaje o estudio y la organización de los aprendizajes, en el marco de lo dispuesto en la Ley Orgánica de Educación Superior [10].

- **Reglamento de las Tutorías Académicas**

El reglamento contiene las normas que regulan el sistema tutorial de la Universidad Técnica de Ambato pensado como un proceso de

acompañamiento durante la formación de los estudiantes por parte de los profesores tutores en el ámbito académico y de formación integral [11].

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar una Aplicación Móvil usando el Framework Flutter para el control de Tutorías Académicas del Sistema Integrado de la Universidad Técnica de Ambato.

1.3.2. Objetivos específicos

- Identificar los requerimientos y las nuevas funcionalidades que tendrá el Módulo de Tutorías para mejorar la interacción con los docentes y estudiantes.
- Investigar el uso del Framework Flutter para el desarrollo de aplicaciones móviles.
- Implementar la aplicación móvil para el control de Tutorías Académicas del Sistema Integrado de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial aplicando el Framework Flutter.

CAPÍTULO II

METODOLOGÍA

2.1. Materiales

Para el presente proyecto, se usó una entrevista al Director Académico y al analista de actividades académicas y dos encuestas para docentes y estudiantes de la Universidad Técnica de Ambato, con el fin de analizar los datos relevantes para el desarrollo de la Aplicación Móvil.

Entrevista aplicada al Director Académico y al analista de actividades académicas de la Universidad Técnica de Ambato durante una reunión vía Zoom.

- **Pregunta 1:** ¿Existe algún reglamento sobre las Tutorías Académicas en la universidad que se deba seguir?
- **Pregunta 2:** ¿Cree usted que sería de gran utilidad para los docentes y estudiantes contar con una Aplicación Móvil que les permita interactuar de una manera más sencilla, práctica y rápida con las Tutorías Académicas?
- **Pregunta 3:** ¿Acorde a las funcionalidades que se le mencionará a continuación, indique cuáles son las más relevantes para los estudiantes si la Aplicación Móvil pudiera contar con las mismas?
 - Solicitar Tutorías Académicas que se registren sin necesidad de que el docente lo haga.
 - Validar las Tutorías Académicas.
 - Ver los estados de las Tutorías Académicas.
 - Recibir notificaciones a tiempo para poder validar las tutorías.
- **Pregunta 4:** ¿Acorde a las funcionalidades que se le mencionará a continuación, indique cuáles son las más relevantes para los docentes si la Aplicación Móvil pudiera contar con las mismas?
 - Registrar Tutorías Académicas.
 - Gestionar los estados de las Tutorías Académicas.

- Ver los estados de las Tutorías Académicas.
 - Recibir notificaciones sobre solicitudes de tutorías.
- **Pregunta 5:** ¿Qué aspectos importantes referentes a las Tutorías Académicas deben ser considerados para el desarrollo de la aplicación?

Encuesta sobre las Tutorías Académicas para Docentes de la Universidad Técnica De Ambato

La encuesta tiene el fin de recopilar información sobre un proyecto de investigación para desarrollar una Aplicación Móvil del Módulo de Tutorías Académicas.

Nota: Para la presente encuesta se tomará en cuenta el horario de trabajo del Docente.

1. ¿Con qué frecuencia un estudiante le solicita una tutoría académica sin ser el horario asignado para dar la tutoría académica de la materia solicitada?

- Nunca
- Raramente
- Ocasionalmente
- Frecuentemente
- Muy frecuentemente

2. ¿Con qué periodicidad usted ayuda a los estudiantes con Tutorías Académicas?

- Nunca
- Raramente

- Ocasionalmente (Según el horario de Tutorías asignado)
- Casi todos los días (Lunes a Viernes)
- Todos los días (Lunes a Viernes)

3. ¿Con qué frecuencia les recuerda a los estudiantes que deben validar las tutorías académicas que usted asistió antes de terminar un parcial?

- Nunca
- Raramente
- Ocasionalmente
- Frecuentemente
- Muy frecuentemente

4. ¿Está de acuerdo en contar con una Aplicación Móvil para interactuar de una manera más sencilla, práctica y rápida con las Tutorías Académicas?

- Totalmente en desacuerdo
- En desacuerdo
- Ni de acuerdo, ni en desacuerdo
- De acuerdo
- Totalmente de acuerdo

5. ¿Suponga, de acuerdo a su horario de trabajo de Tutorías Académicas, un estudiante le solicita una tutoría mediante una Aplicación Móvil y usted acepta, esta Tutoría Académica se creará automáticamente sin necesidad de que usted la registre, ¿cree usted que esta funcionalidad en la Aplicación Móvil le será de gran ayuda?

- Totalmente en desacuerdo
- En desacuerdo
- Ni de acuerdo, ni en desacuerdo
- De acuerdo
- Totalmente de acuerdo

6. ¿Cree usted que es importante que el estudiante luego de haber asistido a una Tutoría Académica se le recuerde que debe validarla recibiendo para el efecto una notificación por medio de una Aplicación Móvil?

- Sin importancia
- De poca importancia
- Moderadamente importante
- Importante
- Muy importante

7. ¿Aceptaría proporcionar una Tutoría Académica que solicite un estudiante por medio de una Aplicación Móvil en horas que tenga disponibilidad de acuerdo al horario de clases, aparte de las Tutorías Académicas asignadas? Por ejemplo, si se encuentra libre en las horas de actividades académicas.

- Totalmente en desacuerdo
- En desacuerdo
- Ni de acuerdo, ni en desacuerdo
- De acuerdo
- Totalmente de acuerdo

8. ¿Cree importante que si usted proporcionó Tutorías Académicas a los estudiantes y estos no lo solicitaron por medio de una Aplicación Móvil, le gustaría que existiera una funcionalidad en la aplicación para registrar estas tutorías y que se les notifique a los estudiantes que deben validar dichas tutorías?

- Sin importancia
- De poca importancia
- Moderadamente importante
- Importante
- Muy importante

Encuesta sobre las Tutorías Académicas para Estudiantes de la Universidad Técnica De Ambato

La presente encuesta tiene el fin de recopilar información sobre un proyecto de investigación para desarrollar una Aplicación Móvil del Módulo de Tutorías Académicas.

1. ¿Alguna vez ha solicitado ayuda o ha preguntado a los Docentes sobre un tema de una materia, dónde requería asesoría en el horario asignado?

Si

No

2. ¿Con qué frecuencia pide ayuda a un Docente sobre un tema de una materia dónde necesita aclarar dudas, en las horas de Tutorías Académicas asignadas?

Nunca

Raramente

Ocasionalmente

Frecuentemente

Muy frecuentemente

3. ¿Con qué frecuencia ha solicitado ayuda o ha preguntado a un Docente sobre un tema fuera del horario de Tutorías Académicas de la asignatura?

- Nunca
- Raramente
- Ocasionalmente
- Frecuentemente
- Muy frecuentemente

4. ¿Está de acuerdo en contar con una Aplicación Móvil para interactuar de una manera más sencilla, práctica y rápida con las Tutorías Académicas?

- Totalmente en desacuerdo
- En desacuerdo
- Ni de acuerdo, ni en desacuerdo
- De acuerdo
- Totalmente de acuerdo

5. ¿Está usted de acuerdo que exista una funcionalidad en una Aplicación Móvil que permita solicitar Tutorías Académicas sobre un tema de una asignatura en el horario que el docente tenga disponibilidad?

- Totalmente en desacuerdo
- En desacuerdo

Ni de acuerdo, ni en desacuerdo

De acuerdo

Totalmente de acuerdo

6. ¿Le gustaría que luego de haber asistido a una tutoría solicitada, mediante una Aplicación Móvil se le notifique para que pueda validarla?

Si

No

7. Si usted no validó las tutorías luego de haber asistido, ¿considera importante que la Aplicación Móvil le notifique a tiempo que tiene tutorías pendientes para validar antes de que se termine cada parcial?

Sin importancia

De poca importancia

Moderadamente importante

Importante

Muy importante

2.2. Métodos

2.2.1. Modalidad de investigación

Investigación Bibliográfica

La investigación fue realizada mediante porque se usaron fuentes como artículos científicos, libros, revistas, documentos de fuentes confiables como repositorios de universidades, que servirán de sustento científico para el proyecto visualizando diferentes puntos de vista para llegar a una solución.

Investigación de Campo

En este proyecto se usó la investigación de campo puesto que se tomó como fuente de información las perspectivas de los involucrados y comprometidos para recopilar datos reales, analizarlos y lograr una solución óptima y factible para resolver el problema.

2.2.2. Población y muestra

Población: Para la presente investigación, la población está conformada por los docentes y estudiantes de la Universidad Técnica de Ambato.

Muestra: Para la presente investigación, se usará la técnica de muestreo no probabilístico para seleccionar al azar a una muestra de docentes y una muestra de estudiantes.

Para la muestra de docentes:

Población Docentes	Número	Porcentaje
Docentes tutores	683	100%
Total:	683	100%

Tabla 6: Población de docentes

Elaborado por: Paola Bautista

- Tamaño de la muestra $N = 683$

- Confianza Z = 0,95 -> 1,96
- Error de muestreo e = 0,05
- Desviación estándar s = 0,5

$$n = \frac{(683) (0,5)^2 (1,96)^2}{(683-1)(0,05)^2 + (0,5)^2 (1,96)^2}$$

n = 246 docentes

Para la muestra de estudiantes:

Población Estudiantes	Número	Porcentaje
Nivelación	1991	11.56%
Pregrado	15235	88.44%
Total:	17226	100%

Tabla 7: Población de estudiantes

Elaborado por: Paola Bautista

- Tamaño de la muestra N = 17,226
- Confianza Z = 0,95 -> 1,96
- Error de muestreo e = 0,05
- Desviación estándar s = 0,5

$$n = \frac{(17,226) (0,5)^2 (1,96)^2}{(17,226-1)(0,05)^2 + (0,5)^2 (1,96)^2}$$

n = 376 estudiantes

2.2.3. Recolección de información

Al aplicarse la entrevista y las encuestas a los stakeholders se obtuvo los siguientes resultados.

2.2.3.1. Resultados de la entrevista aplicada al Director Académico y al Analista de Actividades Académicas de la Universidad Técnica de Ambato

- **Pregunta 1:** ¿Existe algún reglamento sobre las Tutorías Académicas que se deba seguir?

Respuesta: Si existe un Reglamento de las Tutorías Académicas y Grupales [11] de la universidad que se encuentra en la página web oficial dentro de los reglamentos internos.

- **Pregunta 2:** ¿Cree usted que sería de gran utilidad para los docentes y estudiantes contar con una Aplicación Móvil que les permita interactuar de una manera más sencilla, práctica y rápida con las Tutorías Académicas?

Respuesta: Sería una herramienta muy útil para poder cumplir con estas funciones académicas tanto para el docente como para el estudiante. Hay un problema referente a las tutorías debido a que docentes explican que, si asisten las mismas, pero no las registran de inmediato y también los reportes que se generan tienen ciertos inconvenientes debido a que en algunos casos los registros de las tutorías no reflejan la realidad por el mismo motivo.

- **Pregunta 3:** ¿Acorde a las funcionalidades que se le mencionará a continuación, indique cuáles son las más relevantes para los estudiantes si la Aplicación Móvil pudiera contar con las mismas?

- Solicitar Tutorías Académicas que se registren sin necesidad de que el docente lo haga.
- Validar las Tutorías Académicas.
- Ver los estados de las Tutorías Académicas.
- Recibir notificaciones a tiempo para poder validar las tutorías.

Respuesta: Las funcionalidades para estudiantes ayudarían a mejorar la interacción entre los docentes y estudiantes con las tutorías además

de resolver los inconvenientes que se mencionaron en la anterior respuesta.

Como recomendación para solicitar las tutorías, que solo se permita en el horario de trabajo de los docentes para evitar molestias.

- **Pregunta 4:** ¿Acorde a las funcionalidades que se le mencionará a continuación, indique cuáles son las más relevantes para los docentes si la Aplicación Móvil pudiera contar con las mismas?
 - Registrar Tutorías Académicas.
 - Gestionar los estados de las Tutorías Académicas.
 - Ver los estados de las Tutorías Académicas.
 - Recibir notificaciones sobre solicitudes de tutorías.

Respuesta: Las funcionalidades para docentes también ayudarían mucho, e inclusive optimiza el cumplimiento de función de las Tutorías Académicas y evitar el problema de los reportes.

- **Pregunta 5:** ¿Qué aspectos importantes referentes a las Tutorías Académicas deben ser considerados para el desarrollo de la aplicación?

Respuesta: Se debe respetar los horarios de trabajo del docente, y considerar en qué materias el estudiante está matriculado para que pueda solicitar tutorías.

También que exista una notificación para los estudiantes al final de cada parcial en caso de que no haya validado.

2.2.3.2. Resultado de las encuestas aplicadas a los docentes

Pregunta 1: ¿Con qué frecuencia un estudiante le solicita una Tutoría Académica sin ser el horario asignado para dar la Tutoría Académica de la materia solicitada?

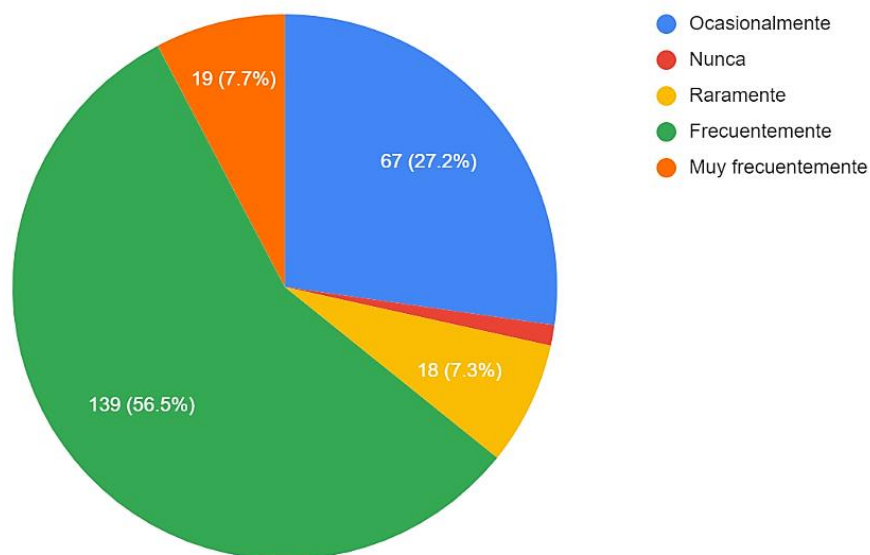


Figura 1: Frecuencia de solicitudes en horarios asignados para las Tutorías Académicas

Elaborado por: Paola Bautista

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

Según los resultados de la figura 1, el 56% de docentes encuestados respondieron que frecuentemente los estudiantes solicitan tutorías en los horarios asignados, el 27.2% ocasionalmente y seguido del 7.7% muy frecuentemente. Indicando que la mayoría de docentes si han asistido tutorías a los estudiantes con frecuencia.

Pregunta 2: ¿Con qué periodicidad usted ayuda a los estudiantes con Tutorías Académicas?

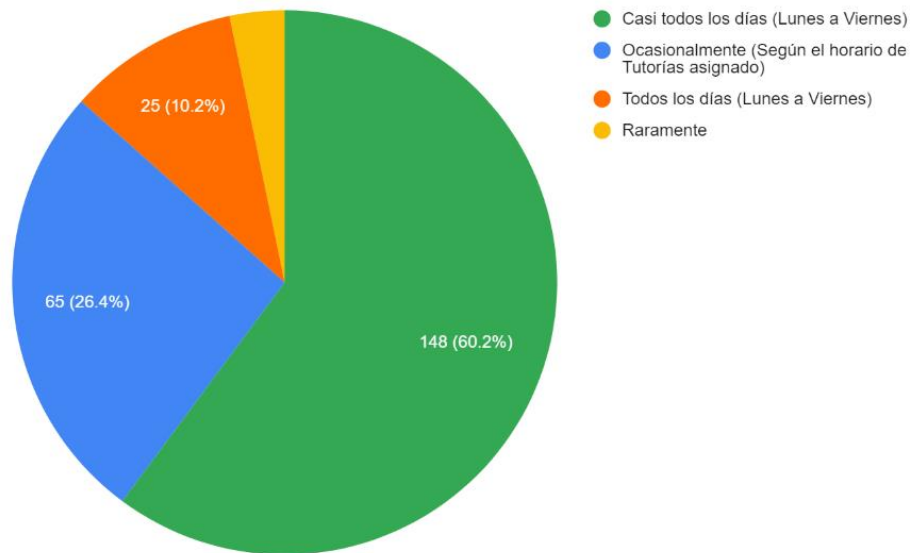


Figura 2: Periodicidad de asistencia de Tutorías Académicas a los estudiantes.

Elaborado por: Paola Bautista

Fuente: Encuesta aplicada

Análisis e interpretación de datos:

De acuerdo con los resultados de la figura 2, el 60.2% siendo la mayoría de docentes encuestados han proporcionado tutorías a los estudiantes casi todos los días de su horario laboral, seguido por el 26.4% que lo han hecho ocasionalmente, por lo que indica que los docentes si ayudan periódicamente a los estudiantes con tutorías académicas.

Pregunta 3: ¿Con qué frecuencia les recuerda a los estudiantes que deben validar las tutorías académicas que usted asistió antes de terminar un parcial?

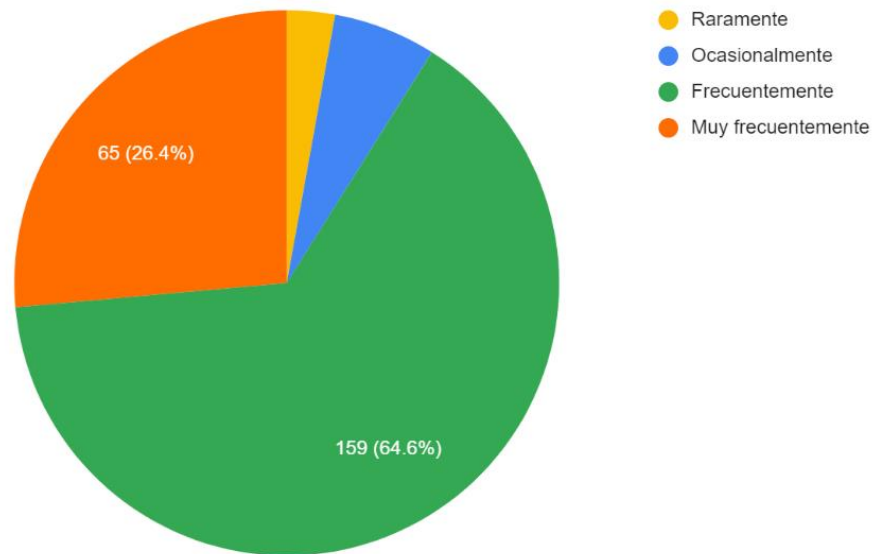


Figura 3: Frecuencia de que el docente le recuerda a los estudiantes de validar las Tutorías Académicas

Elaborado por: Paola Bautista

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

De acuerdo los resultados de la figura 3, el 64.6% de docentes encuestados frecuentemente deben recordarles a los estudiantes que deben validar las tutorías al terminarse cada parcial, así como el 26.4% lo hacen muy frecuentemente.

Pregunta 4: ¿Está de acuerdo en contar con una Aplicación Móvil para interactuar de una manera más sencilla, práctica y rápida con las Tutorías Académicas?

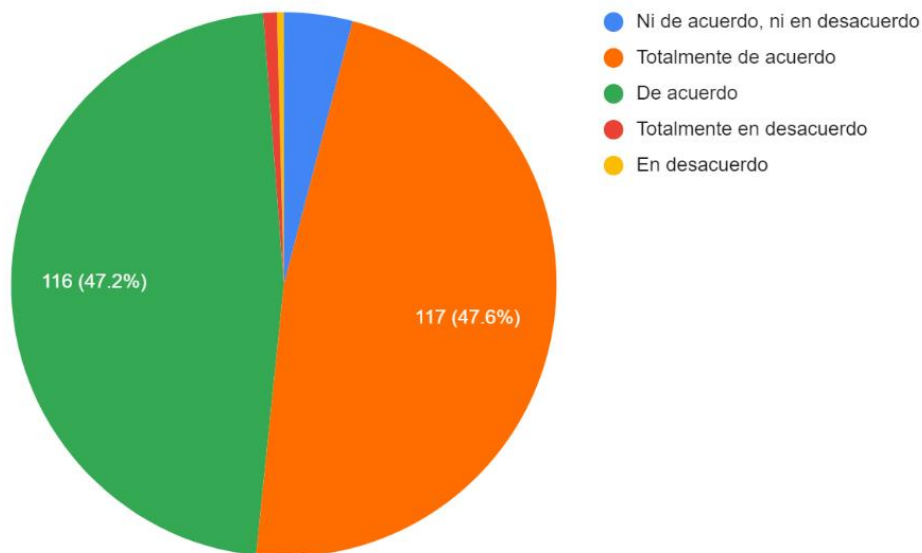


Figura 4: Aprobación de los docentes sobre una App Móvil para mejorar la interactividad de las Tutorías Académicas

Elaborado por: Paola Bautista

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

De acuerdo a los resultados de la figura 4, el 47.6% de docentes que están de acuerdo seguido por el 47.2% que están totalmente de acuerdo, por lo que el 94.8% aprueba que exista una Aplicación Móvil para mejorar la interactividad de las Tutorías Académicas.

Pregunta 5: Suponga, de acuerdo a su horario de trabajo de Tutorías Académicas, un estudiante le solicita una tutoría mediante una Aplicación Móvil y usted acepta, esta Tutoría Académica se creará automáticamente sin necesidad de que usted la registre, ¿cree usted que esta funcionalidad en la Aplicación Móvil le será de gran ayuda?

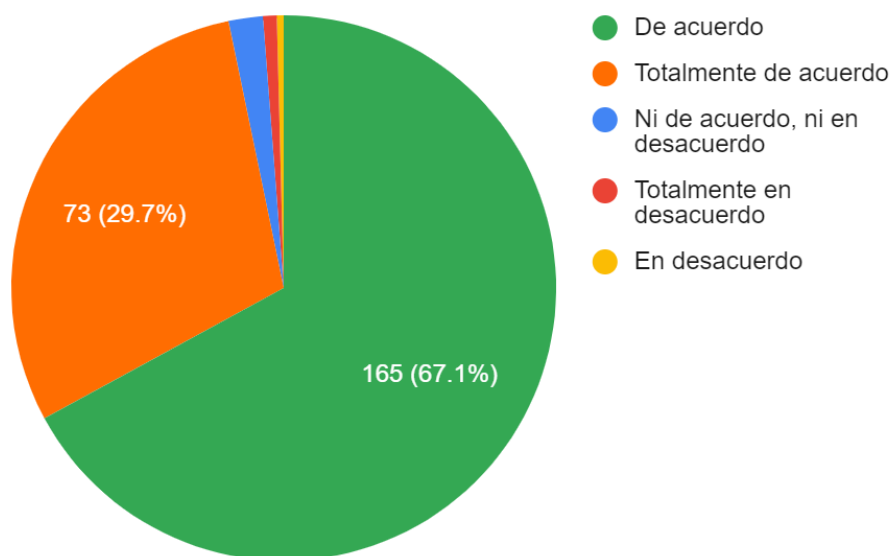


Figura 5: Aprobación de docentes encuestados sobre una app móvil para optimizar el registro de las Tutorías Académicas

Elaborado por: Paola Bautista

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

De acuerdo a los resultados de la figura 5, el 67.1% de docentes que están de acuerdo seguido por el 29.7% que están totalmente de acuerdo, por lo que el 96.8% aprueba que exista una funcionalidad en la Aplicación Móvil para que se registre las tutorías automáticamente luego de que se asistió una tutoría solicitada por la misma app.

Pregunta 6: ¿Cree usted que es importante que el estudiante luego de haber asistido a una Tutoría Académica se le recuerde que debe validarla recibiendo para el efecto una notificación por medio de una Aplicación Móvil?

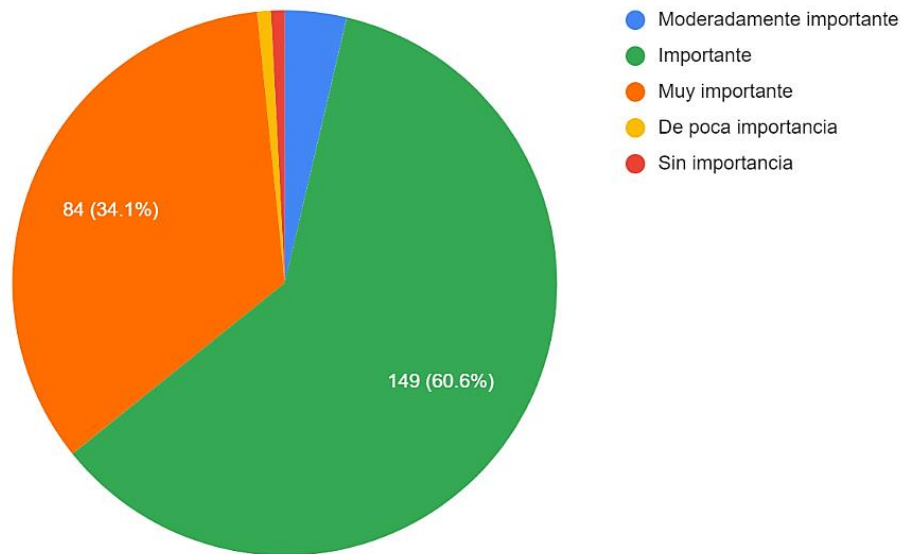


Figura 6: Importancia de que se le notifique al estudiante que debe validar tutorías luego de haberlas asistido

Elaborado por: Paola Bautista

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

De acuerdo con los resultados de la figura 6, el 60.6% de docentes encuestados piensan que es importante que se les notifique a los estudiantes que deben validar las tutorías luego de haberlas asistido, seguido del 34.1% que piensa que es muy importante por lo que la mayoría de docentes les gustaría que exista esta funcionalidad para optimizar este proceso de las Tutorías Académicas.

Pregunta 7: ¿Aceptaría proporcionar una Tutoría Académica que solicite un estudiante por medio de una Aplicación Móvil en horas que tenga disponibilidad de acuerdo al horario de clases, aparte de las Tutorías Académicas asignadas? Por ejemplo, si se encuentra libre en las horas de actividades académicas.

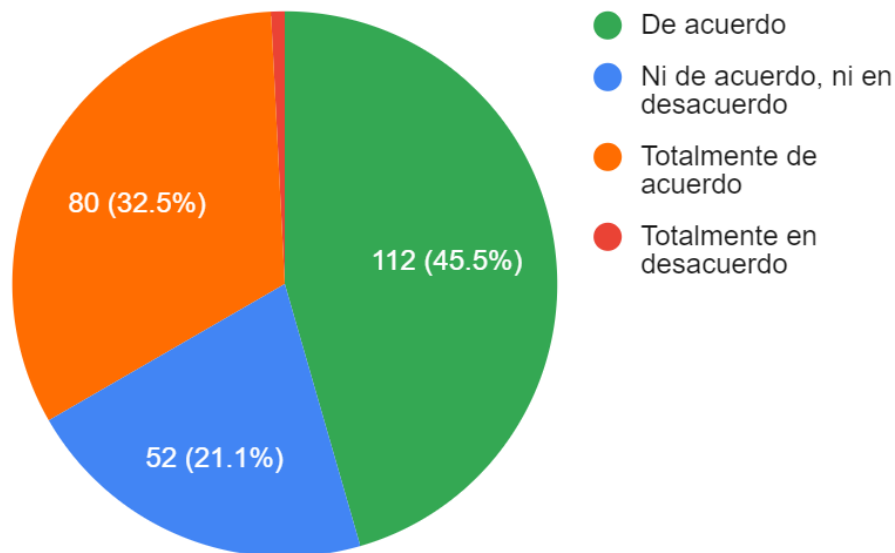


Figura 7: Aceptación de tutorar a los estudiantes en horas de actividades académicas.

Elaborado por: Paola Bautista

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

De acuerdo a los resultados de la figura 7, el 45.5% de docentes encuestados están de acuerdo seguido por el 32.5% que están totalmente de acuerdo, demostrando que la mayoría de docentes aceptan tutorar a los estudiantes en sus horas de actividades académicas.

Pregunta 8: ¿Cree importante que si usted proporcionó Tutorías Académicas a los estudiantes y estos no lo solicitaron por medio de una Aplicación Móvil, le gustaría que existiera una funcionalidad en la aplicación para registrar estas tutorías y que se les notifique a los estudiantes que deben validar dichas tutorías?

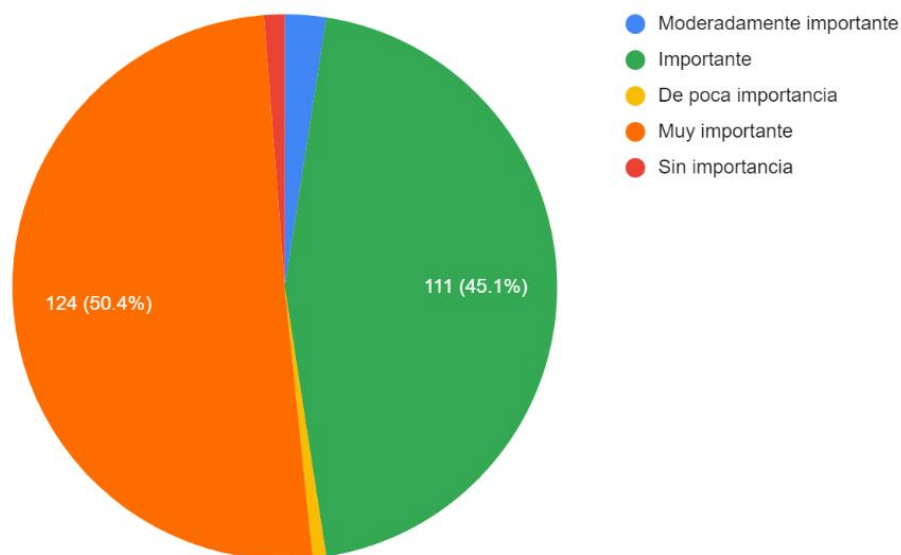


Figura 8: Notificaciones a los estudiantes sobre las Tutorías Académicas por medio de la App Móvil

Elaborado por: Paola Bautista

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

De acuerdo a los resultados de la figura 8, el 50.4% de docentes encuestados creen que es muy importante seguido por el 45.1% que creen que es importante que se les notifique a los estudiantes por medio de una Aplicación Móvil que deben validar las Tutorías Académicas que no fueron solicitadas por la misma app, por lo que piensan que esta funcionalidad les será de gran ayuda.

2.2.3.3. Resultado de las encuestas aplicadas a los estudiantes

Al aplicarse la encuesta a los estudiantes, se obtuvo los siguientes resultados

Pregunta 1: ¿Alguna vez ha solicitado ayuda o ha preguntado a los Docentes sobre un tema de una materia, donde requería asesoría en el horario asignado?

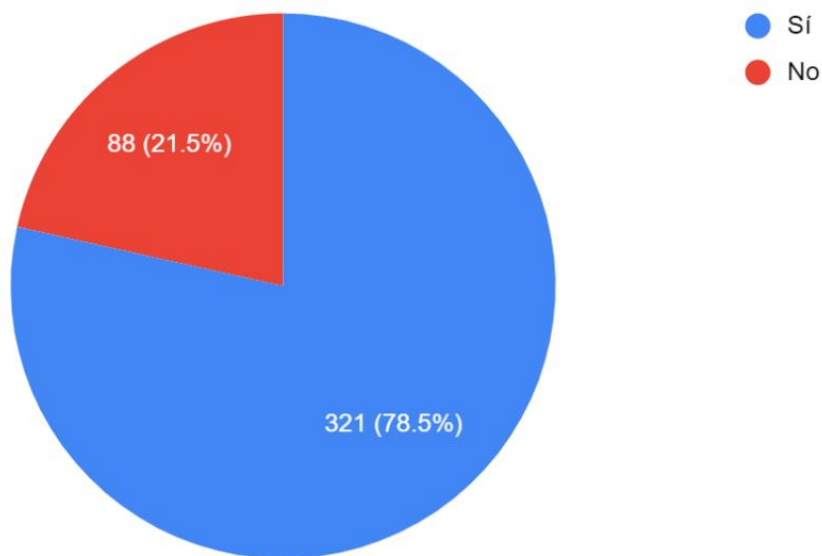


Figura 9: Solicitud de Tutorías Académicas en horas asignadas

Elaborado por: Paola Bautista

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

De acuerdo con los resultados de la figura 9, el 78.5% de estudiantes encuestados han solicitado Tutorías Académicas en las horas asignadas indicando que si existe la necesidad de los docentes para reforzar los conocimientos mediante tutorías.

Pregunta 2: ¿Con qué frecuencia pide ayuda a un Docente sobre un tema de una materia dónde necesita aclarar dudas, en las horas de Tutorías Académicas asignadas?

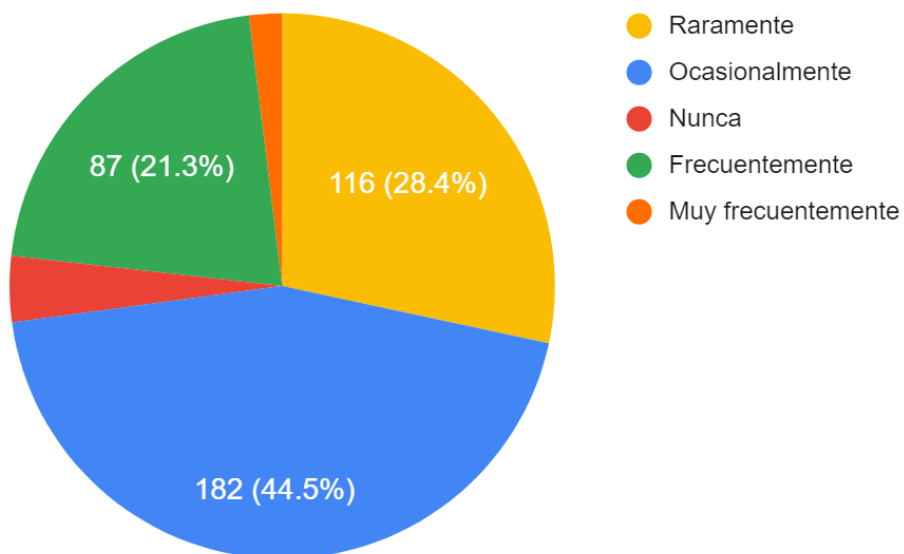


Figura 10: Frecuencia de solicitudes para Tutorías Académicas en horas asignadas

Elaborado por: Paola Bautista

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

De acuerdo a los resultados de la figura 10, el 44.5% de encuestados ocasionalmente han solicitado tutorías en horas asignadas seguido por el 28.4% del encuestados que raramente lo han hecho. Solo el 21.3% ha solicitado frecuentemente y el 3.9% muy frecuentemente, indicando que falta de comunicación entre docentes y estudiantes con las Tutorías Académicas.

Pregunta 3: ¿Con qué frecuencia ha solicitado ayuda o ha preguntado a un Docente sobre un tema fuera del horario de Tutorías Académicas de la asignatura?

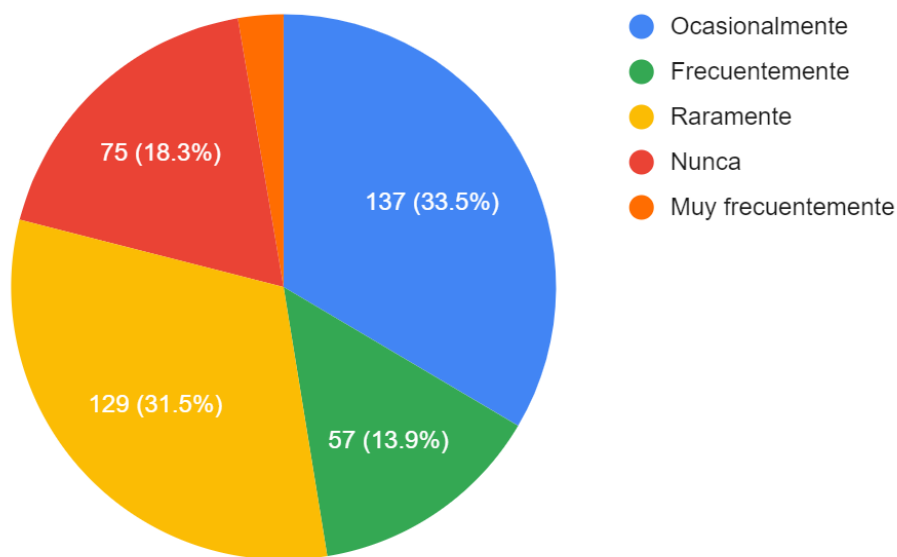


Figura 11: Frecuencia de solicitudes para Tutorías Académicas fuera del horario asignado

Elaborado por: Paola Bautista

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

De acuerdo a los resultados de la figura 11, el 33.5% de estudiantes encuestados han solicitado ocasionalmente tutorías fuera del horario asignado, el 31.5% raramente seguido por el 18.3% que nunca han solicitado. La mayoría de estudiantes han solicitado con poca frecuencia las tutorías académicas fuera del horario asignado, mostrando junto a los resultados de la pregunta 2, que existe falta de comunicación al pedir asistencia a los docentes tutores sobre temas de materias que los estudiantes pueden tener dudas.

Pregunta 4: ¿Está de acuerdo en contar con una Aplicación Móvil para interactuar de una manera más sencilla, práctica y rápida con las Tutorías Académicas?

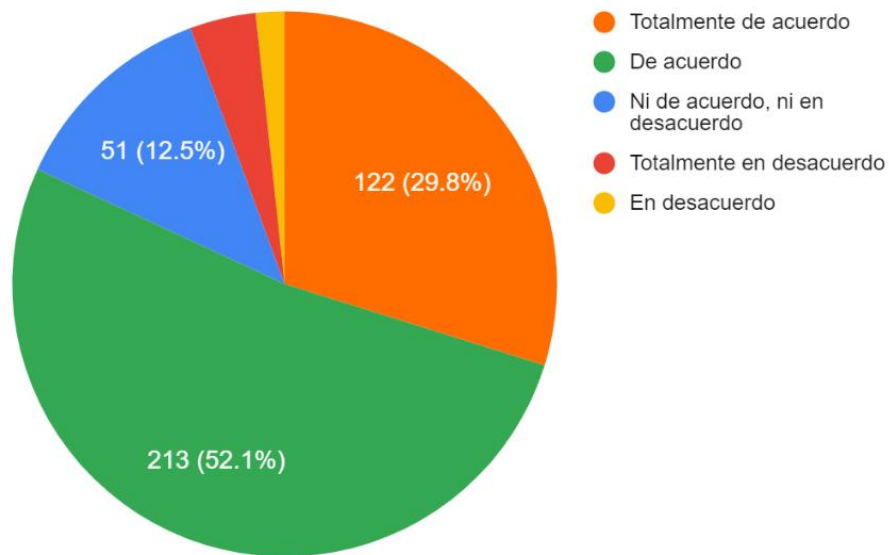


Figura 12: Aprobación de estudiantes encuestados para mejorar la interactividad de las Tutorías Académicas

Elaborado por: Paola Bautista

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

Acorde a los resultados obtenidos de la figura 12, el 52.1% de estudiantes encuestados están de acuerdo, el 29.8% están totalmente de acuerdo, demostrando que la mayoría de estudiantes están de acuerdo con que exista una Aplicación Móvil para mejorar la interactividad con las Tutorías Académicas.

Pregunta 5: ¿Está usted de acuerdo que exista una funcionalidad en una Aplicación Móvil que permita solicitar Tutorías Académicas sobre un tema de una asignatura en el horario que el docente tenga disponibilidad?

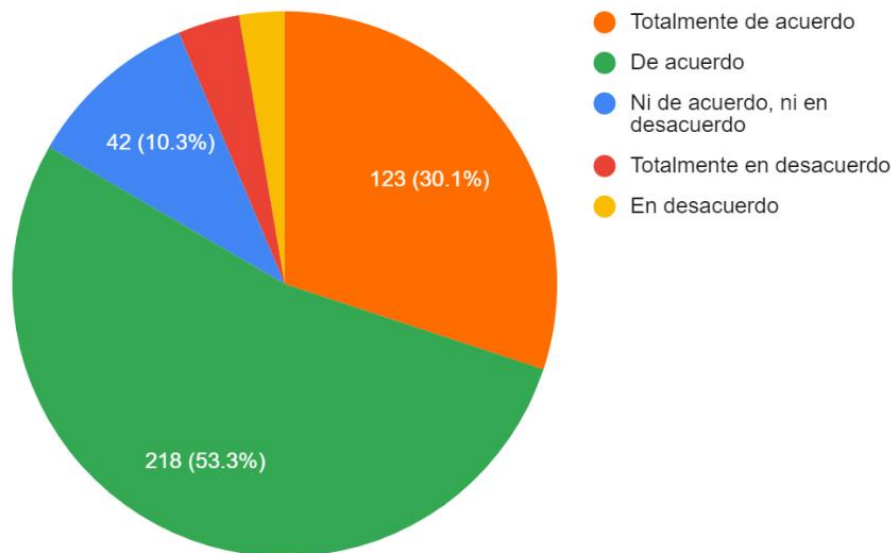


Figura 13: Conformidad de estudiantes encuestados para que la App Móvil pueda solicitar tutorías en cuando el docente tenga disponibilidad.

Elaborado por: Paola Bautista

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

De acuerdo a los resultados de la figura 13, el 53.3% de estudiantes encuestados están de acuerdo y el 30.1% están totalmente de acuerdo, demostrando que la mayoría aprueba que la Aplicación Móvil permita solicitar tutorías en horas que los docentes tengan disponibilidad de acuerdo al horario.

Pregunta 6: ¿Le gustaría que luego de haber asistido a una tutoría solicitada, mediante una Aplicación Móvil se le notifique para que pueda validarla?

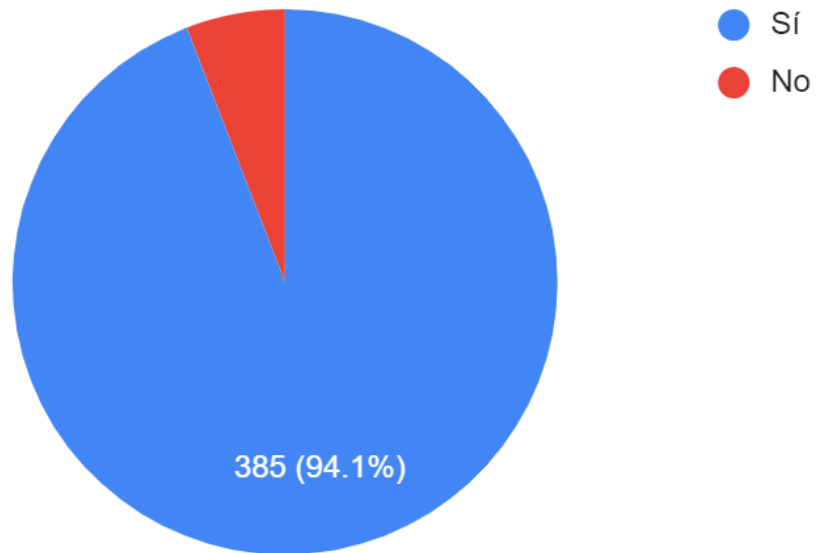


Figura 14: Estudiantes encuestados que quieren notificaciones para validar luego de haber asistido a tutorías.

Elaborado por: Paola Bautista

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

De acuerdo con los resultados de la figura 14, el 94.1% de los estudiantes encuestados si quieren que por medio de la Aplicación Móvil se le notifique luego de haber asistido a una tutoría solicitada por la misma App.

Pregunta 7: Si usted no validó las tutorías luego de haber asistido, ¿considera importante que la Aplicación Móvil le notifique a tiempo que tiene tutorías pendientes para validar antes de que se termine cada parcial?

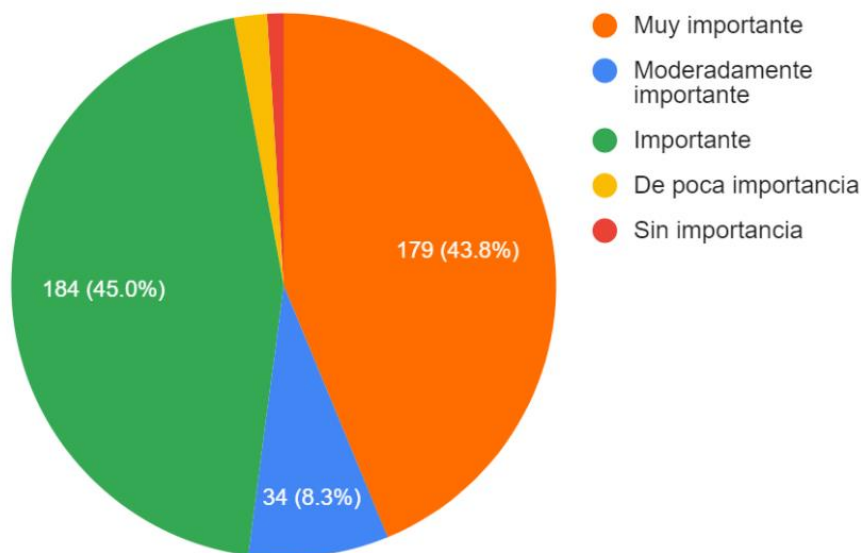


Figura 15: Estudiantes encuestados que quieren notificaciones para validar a tiempo las tutorías antes de terminar cada parcial.

Elaborado por: Paola Bautista

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

De acuerdo a los resultados de la figura 15, El 45% de estudiantes encuestados creen que es importante y el 43.8% creen que es muy importante, que se les notifique a tiempo en cada parcial que les notifique que deben validar las Tutorías Académicas por medio de la Aplicación Móvil.

2.2.4. Procesamiento y análisis de datos

De acuerdo a la entrevista aplicada al Director Académico y al Analista de las Actividades Académicas de la Universidad Técnica de Ambato:

- Aprobaron el desarrollo de la Aplicación Móvil indicando que sería una herramienta muy útil para las Tutorías Académicas, además de que erradicaría los problemas existentes en los reportes puesto que se manejarían datos reales.
- Acorde a las preguntas 3 y 4, las funcionalidades propuestas las consideran apropiadas para poder mejorar las interacciones entre los docentes y estudiantes con las Tutorías Académicas.
- Para las funcionalidades se considerará el Reglamento de las Tutorías Académicas y Grupales de la Universidad Técnica de Ambato.
- Por último, señalaron que es importante respetar los horarios de trabajo de los docentes para evitar molestias e interrupciones fuera de la universidad.

De acuerdo a las encuestas realizadas a los docentes:

- Existe un nivel alto de aprobación para el desarrollo de la Aplicación Móvil para poder mejorar y controlar la interactividad entre los docentes y estudiantes con las Tutorías Académicas.
- Acorde a los resultados obtenidos de las preguntas 6 y 8, las notificaciones por medio de una Aplicación Móvil para que los estudiantes validen las tutorías son muy importantes puesto que ayudará a tener un mejor control e interactividad entre docentes y estudiantes con las Tutorías Académicas.

De acuerdo a las encuestas realizadas a los estudiantes:

- Al igual que los docentes, existe un nivel alto de aprobación por parte de los estudiantes para el desarrollo de la Aplicación Móvil para poder mejorar y controlar la interactividad entre los docentes y estudiantes con las Tutorías Académicas.
- La mayoría de estudiantes encuestados han solicitado tutorías con poca frecuencia ya sea en horario asignado o fuera del horario asignado por lo que indica que existe poca interactividad entre docentes y estudiantes para las

Tutorías Académicas sea por falta de interés de los estudiantes o porque no existe una buena comunicación con los docentes.

- La mayoría de estudiantes creen que les será de gran ayuda que se les notifique a tiempo por medio de la Aplicación Móvil que deben validar las tutorías en cada parcial o al finalizar una tutoría solicitada.

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

3.1. Análisis y discusión de resultados

Acorde a los requerimientos establecidos se pudo desarrollar las funcionalidades necesarias para poder cumplirlas. De esta manera fue importante el análisis de los distintos componentes que fueron parte importante del presente proyecto.

3.1.1. Flutter

Flutter tiene su propio ecosistema netamente para aplicaciones móviles, puesto que se usa el lenguaje de programación Dart para crearlas y se basa en su propio conjunto de estándares que permite aprovechar las funcionalidades nativas del dispositivo móvil y que además permite por medio de un motor de renderizado personalizado mostrar una interfaz de usuario atractivo.

El desarrollo está soportado en Widgets y se basa en la programación orientada a objetos puesto que los widgets pueden estar anidados entre ellos, cuentan con sus propiedades y permiten editarlos. Además de que soporta los cambios de los estados por medio de las funciones integradas.

Características de Flutter que benefician a la Aplicación Móvil:

Características	Flutter
Lenguaje de desarrollo	Dart
Elementos de interfaz	Widgets
Sin conexión a internet	Si
Rendimiento móvil	Excelente por su motor renderizado personalizado
Tasa de refresco de vistas	60 FPS
Compilación	Lenguaje compilado anticipadamente (AOT)

Tabla 8: Características de Flutter

Elaborado por: Paola Bautista

Importancia del uso de Flutter para el desarrollo de la Aplicación Móvil

La investigación sobre Flutter fue uno de los aspectos más importantes debido a que permitió tener una perspectiva más clara y también comprobar y afirmar porqué es el framework más óptimo para el desarrollo del presente proyecto.

Criterio	Descripción
Facilidad de desarrollo	Permite realizar un desarrollo rápido ya que optimiza el tiempo y permite una personalización completa por lo que se adaptó correctamente a las necesidades para el desarrollo del presente proyecto.
Legibilidad del código fuente	Debido a una de sus principales características y experiencia de desarrollo, es fácil comprender el código debido a que permite codificar en base a widgets y además está diseñado en base a una arquitectura por capas.
Herramientas de depuración	Flutter cuenta con una herramienta útil de depuración que permite navegar los árboles de widgets que se forman durante el desarrollo.
Herramientas de generación de código	Las librerías integradas en Flutter permite que el comportamiento de la aplicación sea nativo aprovechando las características del dispositivo móvil.
Comunicación de soporte	Existe una variedad de documentación que permite solucionar los inconvenientes que pueda presentarse. Además, cuenta con el personal adecuado que también ayuda a resolver las dudas.

Tabla 9: Importancia del uso del Framework Flutter

Elaborado por: Paola Bautista

3.1.2. Arquitectura de la aplicación móvil

Desarrollar una aplicación móvil siguiendo un patrón de diseño o arquitectura permite tener un nivel alto de estructura, interacción y funcionamiento como resultado puesto que es una técnica que permite evitar riesgos y corregir errores

3.1.2.1. Arquitectura MVVM

La arquitectura Modelo-Vista-VistaModelo (Model-View-ViewModel) permite separar al máximo las vistas o interfaces de usuario con la lógica de negocio del software. Por lo tanto, facilitará las pruebas, escalabilidad y mantenimiento del proyecto.

- **Modelo:** Representa la información que se manejará en la Aplicación Móvil por lo que conforma la capa de datos.
- **Vista:** Las vistas representan las interfaces de usuario representando la parte visual y las interacciones prometidas de las funcionalidades de la Aplicación Móvil. Entre estas serán las animaciones, eventos, etc.
- **VistaModelo:** Este componente es el intermediario entre la capa de Modelo y la capa de la Vista debido a que permite la comunicación entre la lógica de negocio y los datos, esto quiere decir que las ejecuciones de las peticiones provenientes de las funcionalidades realizadas por los docentes y estudiantes.

Referente al uso de la arquitectura MVVM en Flutter, se implementa mediante las vistas creadas se instancia la clase *ChangeNotifier* y con ayuda de la librería *Provider*, se ejecutarán las funcionalidades requeridas sin problemas

3.1.3. Metodología de desarrollo

En el año 2001 surgió el término ágil debido a que es un modelo del ciclo de desarrollo en iteraciones, por lo tanto, las metodologías ágiles de desarrollo permiten estructurar, planificar y controlar el desarrollo de software.

3.1.3.1. Metodologías ágiles más usadas para aplicaciones móviles

Las Metodologías Ágiles [12], más conocidas son:

Criterio	Scrum	XP	Mobile - D
Tamaño de proyectos	Grandes, medianos y pequeños.	Medianos y pequeños.	Pequeños.
Estilo de desarrollo	Permite mejoras continuas.	Acepta cambios durante las iteraciones.	Acepta cambios durante los días de desarrollo.
Trabajo en equipo	Se puede asignar roles en trabajo de equipo.	Equipo de trabajo por pares.	Grupos pequeños de trabajo.
Documentación	<ul style="list-style-type: none"> • Product backlog • Sprint backlog • Burndown chart • Definition of done • Definition of ready 	<ul style="list-style-type: none"> • Historias de usuario • Tarjetas CRC 	<ul style="list-style-type: none"> • Storycards
Tiempo de producción	Cortos	Largos	Cortos

Tabla 10: Metodologías Ágiles

Elaborado por: Paola Bautista

Acorde a lo investigado, Mobile – D se adapta a las necesidades del presente proyecto debido a que se basa en las metodologías Extreme Programming, Crystal Methodologies y Rational Unified Process, tomando lo mejor de cada una de ellas para la creación de aplicaciones móviles. Consta de cinco fases y cada uno tiene un número de etapas, tareas y prácticas, por lo que sirve para poder cumplir los requerimientos y por lo tanto las funcionalidades establecidas a implementarse en la Aplicación Móvil para las Tutorías Académicas.

3.1.3.2. Metodología Mobile – D

Las fases a cumplirse acorde a la Aplicación Móvil para las Tutorías Académicas son:



Figura 16: Fases de la Metodología Mobile – D

Elaborado por: Paola Bautista

Fase 1: Exploración

- Definir a los interesados del proyecto.
- Definir el alcance de la Aplicación Móvil para el control de las Tutorías Académicas.
- Definición del proyecto.

Fase 2: Inicialización

- Análisis de los requerimientos.
- Arquitectura del software.
- Determinación de las funcionalidades.
- Interfaces de usuario.
- Diagrama de casos de uso.

Fase 3: Producción

- Días de Planificación: análisis, mejoras y priorización de requisitos.
- Días Laborables: implementación de funcionalidades guiado por pruebas de software.
- Días de Liberación: Pruebas de funcionamiento con el tutor.

Fase 4: Estabilización

- Se realiza tareas similares a la fase de Producción para mejorar la Aplicación Móvil en caso de ser necesario.

- Funcionamiento de toda la aplicación.
- Documentación del proyecto.

Fase 5: Pruebas del sistema

- Pruebas de la Aplicación Móvil con el fin de que sea estable y funcional para los involucrados y comprometidos.

3.2. Desarrollo de la propuesta

Para el presente proyecto, se desarrolló aplicando las cinco fases de la Metodología Mobile – D.

3.2.1. Fase I: Exploración

En la primera fase tiene como resultado la definición de stakeholders, el alcance y la determinación del proyecto.

3.2.1.1. Definición de los stakeholders

- **Empresa:** Universidad Técnica de Ambato.
- **Usuarios de la aplicación:** Docentes y estudiantes de la Universidad Técnica de Ambato.
- **Desarrollador:** Autor del presente proyecto.

3.2.1.2. Definición del alcance

- Las funcionalidades que tiene la Aplicación Móvil son acordes a los requisitos establecidos y los resultados de la información obtenida por medio de la entrevista y de las encuestas realizadas a los docentes y estudiantes.
- Los datos se obtendrán por medio de una API REST con el fin de mantener la confidencialidad y la integridad de los datos de los stakeholders.
- La Aplicación Móvil tiene las funcionalidades acordes a los roles, con el fin de cumplir con lo establecido:
 - **Funcionalidades para Docentes:** Registrar tutorías, Gestión de tutorías, Ver estados de tutorías y notificaciones.

- **Funcionalidades para estudiantes:** Solicitar tutorías, Validar tutorías, Ver estados de tutorías y notificaciones.

- **Limitaciones:**

Las limitaciones que tiene la Aplicación Móvil son las siguientes:

- El docente y estudiante debe tener acceso a internet para poder usar la aplicación.
- Los estudiantes solo podrán solicitar tutorías en los horarios disponibles y permitidos acorde al horario del docente tutor.
- Los usuarios que usen dispositivos móviles Android, deben tener versiones mayores o igual a Android 7.0, mientras que para IOS se permite mayores o igual a la versión 9.0 por motivo de compatibilidad.

3.2.1.3. Establecimiento del proyecto

Las herramientas que se usaron en el presente proyecto se describen a continuación:

- **Visual Studio Code**

Editor de código compatible con el lenguaje de programación Dart.

- **Dart**

Lenguaje de programación que permite el desarrollo orientado a widgets en base al Framework Flutter.

- **SQL Server**

Sistema Gestor de Base de datos (SGBD) que permite una amplia variedad de procesamiento de transacciones con el fin de acceder a datos de forma rápida y estructurada.

- **API REST**

Es un conjunto de reglas que permiten de manera flexible y ligera la integración de aplicaciones y dispositivos con el fin de conectar y comunicarse entre sí, pero manteniendo separado el cliente con el servidor. La comunicación se realiza por medio de solicitudes HTTP para insertar, actualizar o eliminar registros de las bases de datos.

3.2.2. Fase II: Inicialización

En la segunda fase como resultado se obtiene el diseño de la Aplicación Móvil.

3.2.2.1. Configuración del ambiente de desarrollo

La configuración permitió montar el ambiente de trabajo por medio de la instalación y configuración de las herramientas necesarias para el desarrollo de la Aplicación Móvil.

- **Tipo de aplicación:** nativa.
- **Framework:** Flutter.
- **Framework para la API RESTful:** ASP.NET Core 5.
- **Sistema Gestor de Base de Datos:** SQL Server.

Preparación del Ambiente

Las instalaciones y configuraciones van acorde a las herramientas que fueron usadas para el desarrollo:

- SQL Server.
- Framework Flutter:
 - Visual Studio Code.
 - Lenguaje de programación Dart.
 - Android Studio.
- Framework ASP.NET Core 5.
 - Visual Studio 2019.
 - Componente de desarrollo ASP.NET Core.
 - Paquetes NuGet.
- Internet Information Services (IIS).

3.2.2.2. Planificación de fases

Fase	Iteración	Descripción
Exploración	Iteración 0	<ul style="list-style-type: none"> Definición de los stakeholders. Definición del alcance. Definición del proyecto.
Inicialización	Iteración 1	<ul style="list-style-type: none"> Configuración del ambiente de desarrollo.
	Iteración 2	<ul style="list-style-type: none"> Diseño de la base de datos. Análisis de los requisitos iniciales.
Producción	Iteración 3	<ul style="list-style-type: none"> Análisis e implementación de la funcionalidad de Inicio de Sesión para docentes y estudiantes. Creación y actualización de la storycard. Pruebas de funcionalidad.
	Iteración 4	<ul style="list-style-type: none"> Análisis e implementación de la funcionalidad de Solicitar Tutorías Académicas para los estudiantes. Creación y actualización de la storycard. Pruebas de funcionalidad.
	Iteración 5	<ul style="list-style-type: none"> Análisis e implementación de la funcionalidad de Validar las Tutorías Académicas para los estudiantes. Creación y actualización de la storycard. Pruebas de funcionalidad.

	Iteración 6	<ul style="list-style-type: none"> • Análisis e implementación de la funcionalidad de Ver estados de las Tutorías Académicas para los estudiantes. • Creación y actualización de la storycard. • Pruebas de funcionalidad.
	Iteración 7	<ul style="list-style-type: none"> • Análisis e implementación de la funcionalidad de Registro de Tutorías Académicas para los docentes. • Creación y actualización de la storycard. • Pruebas de funcionalidad.
	Iteración 8	<ul style="list-style-type: none"> • Análisis e implementación de la funcionalidad de Gestión de estados de las Tutorías Académicas para los docentes. • Creación y actualización de la storycard. • Pruebas de funcionalidad.
	Iteración 9	<ul style="list-style-type: none"> • Análisis e implementación de la funcionalidad de Ver estados de las Tutorías Académicas para los docentes. • Creación y actualización de la storycard. • Pruebas de funcionalidad.
	Iteración 10	<ul style="list-style-type: none"> • Análisis e implementación de las notificaciones para docentes y estudiantes

		<p>dependiendo de la funcionalidad.</p> <ul style="list-style-type: none"> • Creación y actualización de la storycard. • Pruebas de funcionalidad.
Estabilización	Iteración 11	<ul style="list-style-type: none"> • Ajustes del diseño de la base de datos. • Ajustes de widgets en las interfaces de la aplicación.
Pruebas del Sistema	Iteración 12	<ul style="list-style-type: none"> • Pruebas de la Aplicación Móvil acorde a los requerimientos y funcionalidades establecidas.

Tabla 11: Planificación de fases

Elaborado por: Paola Bautista

3.2.2.3. Diseño de la base de datos

Modelo Conceptual

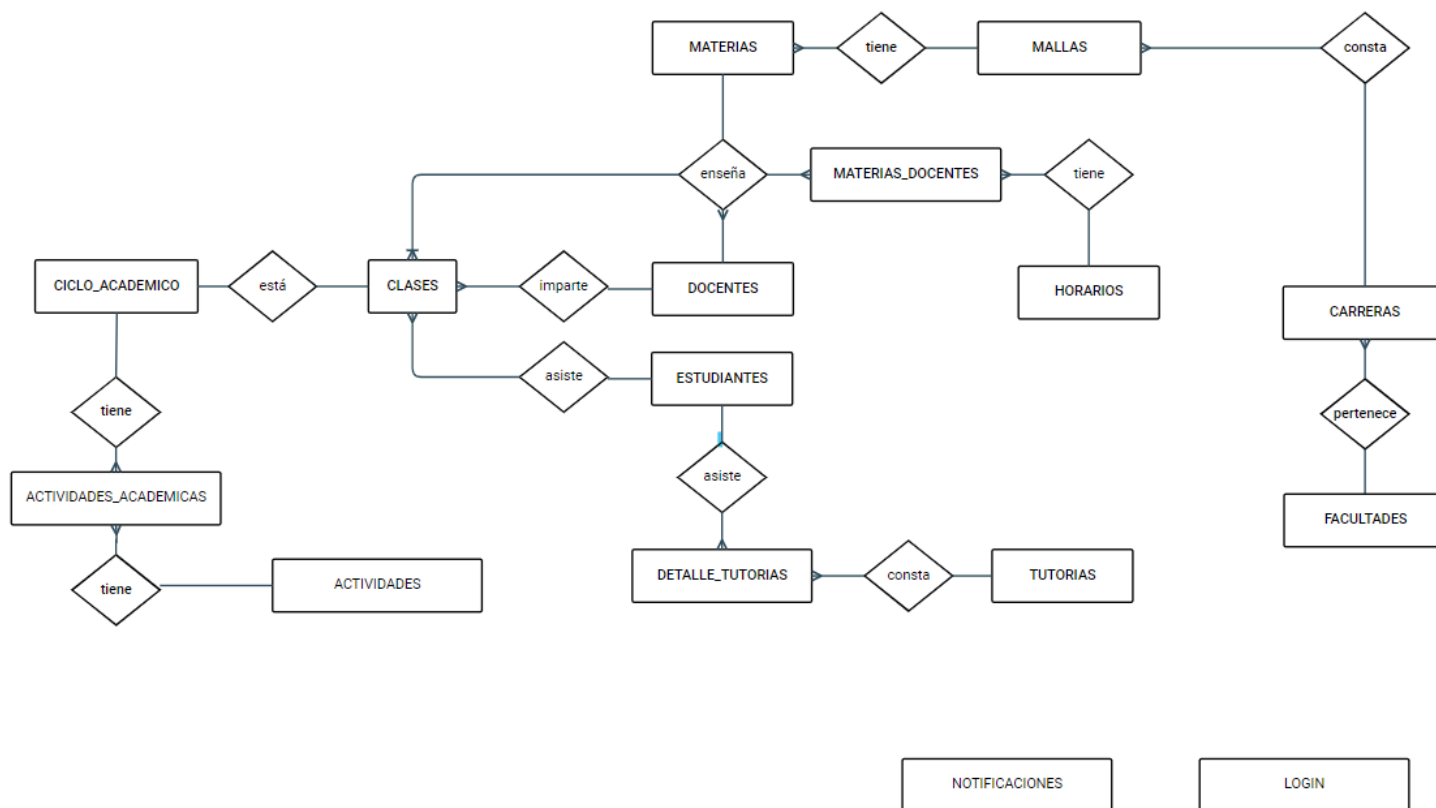


Figura 17: Modelo Conceptual

Elaborado por: Paola Bautista

Modelo Relacional

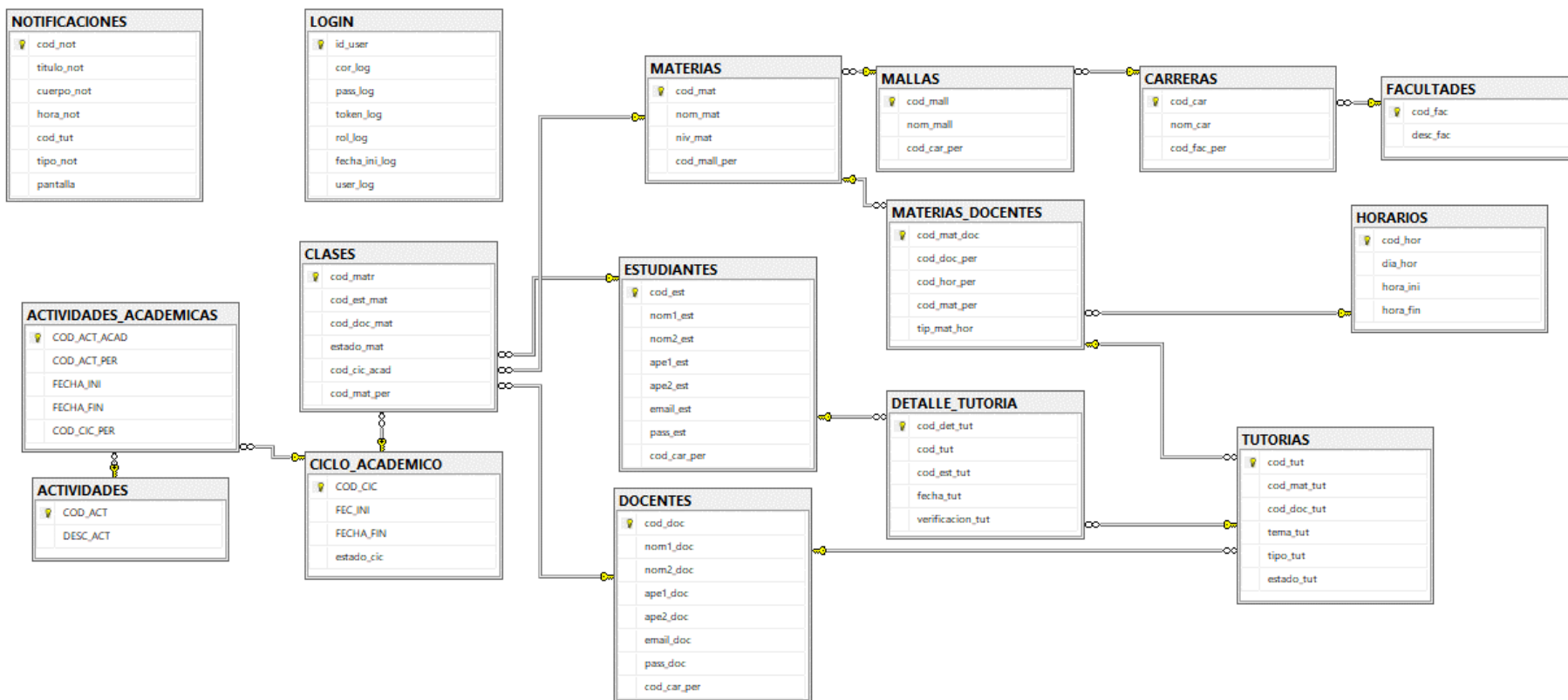


Figura 18: Modelo Relacional

Elaborado por: Paola Bautista

3.2.2.4. Arquitectura de la aplicación móvil

En la arquitectura propuesta, los usuarios (docentes y estudiantes) por medio de la Aplicación Móvil pueden ejecutar las funcionalidades para las Tutorías Académicas. Dependiendo de la funcionalidad solicitada por el usuario como muestra la figura 19, pasa por internet la petición HTTP a la API que se encarga de comunicarse con la base de datos y ejecuta las acciones para que finalmente retorne el resultado deseado.

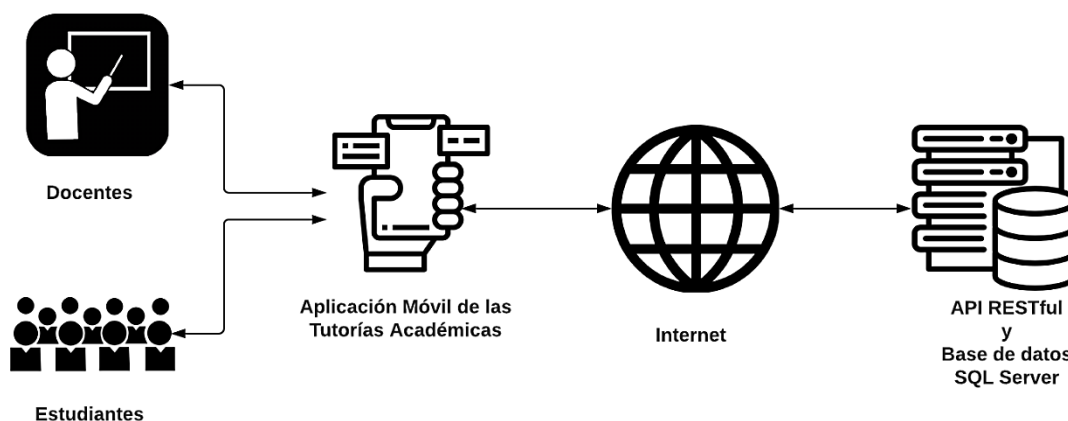


Figura 19: Arquitectura física de la Aplicación Móvil

Elaborado por: Paola Bautista

3.2.2.5. Análisis de los requisitos iniciales

Requerimientos acordes a los roles establecidos para las funcionalidades en la Aplicación Móvil.

Estudiantes:

- El estudiante podrá solicitar Tutorías Académicas sobre un tema de una materia en el que esté matriculado o cursando.
- El estudiante solo podrá solicitar tutorías a los docentes siempre y cuando esté dentro de los horarios asignados y permitidos para las Tutorías Académicas.
- El estudiante podrá ver los estados de las tutorías.
- El estudiante podrá registrar las tutorías solicitadas luego de haber asistido o cuando esté finalizando cada parcial.
- El estudiante recibirá las notificaciones acordes a las funcionalidades ejecutadas.

Docentes:

- El docente podrá rechazar las solicitudes de Tutorías Académicas por parte de los estudiantes dependiendo de su decisión en las horas de Actividades Académicas.
- El docente podrá ver los estados de las tutorías.
- El docente podrá crear registros de tutorías en caso de que el estudiante no haya solicitado una tutoría, pero el docente si le proporcionó ayuda sobre un tema de una materia.
- El docente recibirá notificaciones en caso de que el estudiante le solicite una Tutoría Académica o para indicarle al final de cada parcial que hay registro de tutorías pendientes de validar por parte de los estudiantes.
- El docente recibirá notificaciones en los horarios asignados para las Tutorías Académicas con el fin de no interrumpir o interferir en los horarios de clase o actividades del docente fuera de la Universidad.

3.2.2.6. Funcionalidades establecidas

Acorde a los requerimientos establecidos y los resultados obtenidos por la entrevista al Director Académico y al Analista de Actividades Académicas y las encuestas aplicadas a los docentes y estudiantes de la Universidad Técnica de Ambato, se pudo establecer las funcionalidades que tendrá la Aplicación Móvil.

Funcionalidades que el estudiante puede interactuar por medio de la Aplicación Móvil con los docentes y las Tutorías Académicas

Funcionalidad	Opción	Actividad
Inicio de Sesión	Correo	El estudiante debe insertar su correo y contraseña institucional.
	Contraseña	
Solicitar Tutorías Académicas	Materia	El estudiante tiene las opciones de las materias para escoger solo las que tenga como estado “MATRICULADO”.
	Horario	El estudiante puede seleccionar un horario asignado para tutorías de la materia seleccionada o en los horarios de las horas de actividades académicas del docente.
	Tema	El estudiante tiene la libertad de escribir el tema sobre la materia escogida que desee la asistencia del docente seleccionado.
Observación		
<p>Según el artículo 14 en el Reglamento de Tutorías Académicas y Grupales de la Universidad Técnica de Ambato [11], el docente debe proporcionar refuerzo académico a los estudiantes que lo requieran.</p> <p>De esta manera, con la funcionalidad le permitirá cumplir dicha función.</p> <p>Un aspecto importante de esta funcionalidad es que, si el docente acepta la solicitud, automáticamente se creará el registro de la Tutoría Académica sin necesidad de que el docente lo crea.</p>		
Ver estados de las Tutorías Académicas.	Materia	El estudiante puede ver los estados de las tutorías. Los estados de las tutorías serán:

		<ul style="list-style-type: none"> • Rechazada (Si el docente rechaza la solicitud del estudiante). • Pendiente (Cuando se emitió la solicitud y está en curso). • Finalizada (Cuando ya finalizó la tutoría).
Validar Tutorías Académicas		Permite al estudiante validar las Tutorías Académicas solicitadas por el mismo o las que el docente registró en el supuesto de que haya acontecido este caso.
Observación		
Según el artículo 17 en el Reglamento de Tutorías Académicas y Grupales de la Universidad Técnica de Ambato [11], tiene la obligación de validar el cumplimiento de actividades desarrolladas en la tutoría.		
Notificaciones		<p>Acorde a los resultados obtenidos en la entrevista y las encuestas aplicadas:</p> <ul style="list-style-type: none"> ▪ El estudiante recibirá una notificación cuando el docente rechace su solicitud. ▪ El estudiante recibirá una notificación Luego de haber asistido a una Tutoría Académica con el fin de que valide este registro. ▪ El estudiante recibirá una notificación una semana antes de terminar cada parcial en caso de que tenga validaciones de tutorías pendientes.

Observación

El estudiante solo podrá realizar solicitudes en horarios asignados para tutorías con el fin de que no interrumpir o interferir en horarios de clase y en las actividades del docente fuera de la Universidad.

Tabla 12: Funcionalidades para los estudiantes

Elaborado por: Paola Bautista

Funcionalidades que el docente puede interactuar por medio de la Aplicación Móvil con los estudiantes y las Tutorías Académicas

Funcionalidad	Opción	Actividad
Inicio de sesión	Correo	El docente debe insertar su correo y contraseña institucional.
	Contraseña	
Gestión o ver estado de Tutorías solicitadas	Materia	El docente puede seleccionar la materia para poder ver las solicitudes enviadas por los estudiantes.
	Solicitudes	El docente puede editar el estado de una solicitud, siempre y cuando este sea en horas de actividades académicas y que se encuentre como pendiente.

Observación

Acorde a la entrevista y encuestas aplicadas a los stakeholders, el docente puede rechazar una tutoría en caso de que el estudiante haya solicitado en un horario de actividad académica y no en el horario asignado. No puede rechazar en el horario asignado debido a que en el artículo 14 del Reglamento de Tutorías Académicas y Grupales de la Universidad Técnica de Ambato [11], indica que debe cumplir con el desarrollo de la tutoría en las horas asignadas semanalmente en su distributivo de trabajo a este fin.

Registros de Tutorías Académicas	Materia	El docente selecciona la materia que desee registrar una tutoría proporcionada a los estudiantes.
	Tema	El tema será acorde a lo solicitado por el estudiante y la materia tutorada.
	Horario	El horario será el día y hora en el que fue proporcionada la tutoría.
	Listado de estudiantes	El docente puede seleccionar a los estudiantes poder registrar la tutoría proporcionada.
Observación		
<p>Según el Instructivo del Sistema Integrado de Información de la Universidad Técnica de Ambato "SI-UTA" [13], el docente puede registrar las tutorías académicas.</p> <p>Acorde a los resultados obtenidos de la encuesta aplicada para docentes de la pregunta 8, el docente podrá crear registros de Tutorías Académicas, en caso de que el docente proporcionó una tutoría a un estudiante, pero este no la solicitó por la Aplicación Móvil.</p>		
Ver los estados de las Tutorías Académicas	Materia	El docente puede seleccionar la materia que desee para ver las validaciones de las Tutorías Académicas del actual parcial.
Notificaciones		<ul style="list-style-type: none"> ▪ Acorde a los resultados obtenidos de las encuestas de los docentes y estudiantes: ▪ Recibirá las notificaciones cuando un estudiante le solicite una Tutoría Académica. ▪ Una semana antes de que se termine cada parcial, le llegará

		una notificación indicando que aún falta estudiantes que validen tutorías académicas.
Observación		
Las notificaciones les llegarán a los docentes en horario asignados para tutorías con el fin de no interferir en sus horarios de clase o en actividades del docente fuera de la Universidad.		

Tabla 13: Funcionalidades para el docente

Elaborado por: Paola Bautista

3.2.2.7. Diagramas de caso de uso

Los diagramas de caso de uso tanto para los docentes y estudiantes, indican la relación e interacción con las funcionalidades que tiene la Aplicación Móvil respectivamente.

Diagrama de caso de uso de Estudiantes

La figura 20 indica las acciones que pueden ejecutar los estudiantes por medio de la Aplicación Móvil.

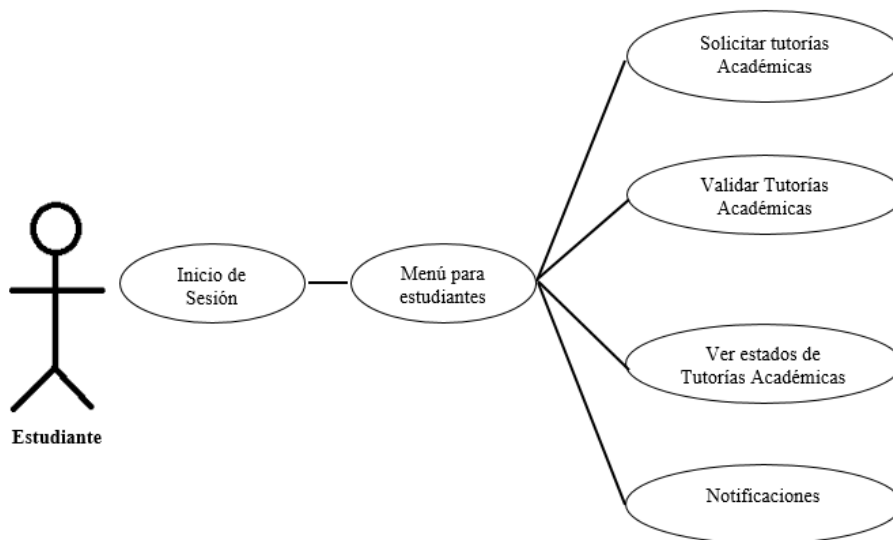


Figura 20: Diagrama de caso de uso de los Estudiantes

Elaborado por: Paola Bautista

Diagrama de caso de uso de Docentes

La figura 21 indica las acciones que pueden ejecutar los docentes por medio de la Aplicación Móvil

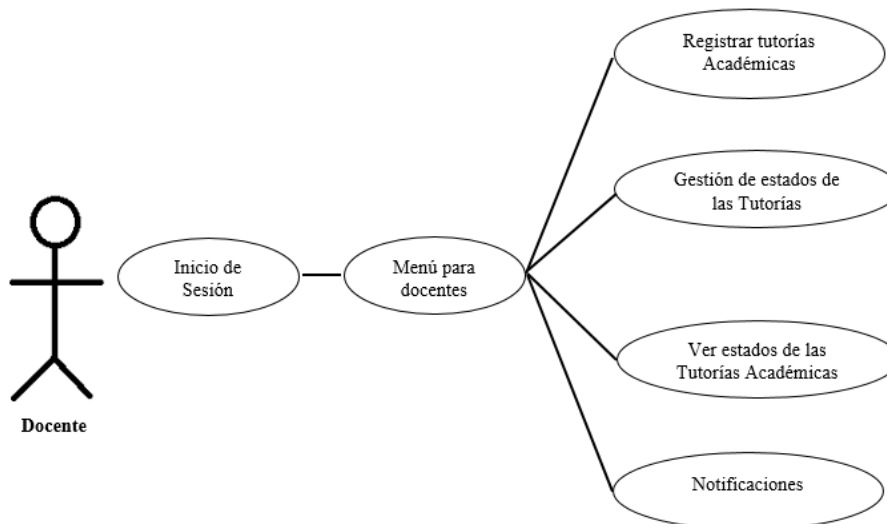


Figura 21: Diagrama de caso de uso de los Docentes

Elaborado por: Paola Bautista

3.2.2.8. Prototipo de la aplicación móvil

Para el presente proyecto se creó un prototipo inicial usando la herramienta Adobe XD para el diseño de las vistas con el propósito de tener una previsualización de las funcionalidades tanto para docentes como para estudiantes.

Inicio de Sesión de docentes y estudiantes

El diseño del Inicio de Sesión muestra como campos de texto de entrada el correo y contraseña con muestra la figura 22.



Figura 22: Inicio de Sesión de usuarios y docentes

Elaborado por: Paola Bautista

Prototipo para Estudiantes

Acorde al rol, se diseñó el prototipo de las vistas para las funcionalidades que el estudiante podrá ejecutar mediante la Aplicación Móvil.

Por cada opción existe una previsualización que indica los distintos elementos que permitirán interactuar al estudiante referente a las Tutorías Académicas.

Menú: En la figura la figura 23, indica el menú para estudiantes.



Figura 23: Menú de estudiantes

Elaborado por: Paola Bautista

Solicitar una tutoría: el estudiante tendrá un diseño como indica la figura 24 para poder solicitar tutorías.



Figura 24: Solicitud de Tutorías

Elaborado por: Paola Bautista

Registrar una tutoría: En la figura 25, indica el diseño de registro de tutorías.



Figura 25: Registro de tutorías

Elaborado por: Paola Bautista

Ver estado de tutorías: En la figura 26, indica el diseño de ver de los estados de tutorías.



Figura 26: Ver estados de tutorías

Elaborado por: Paola Bautista

Prototipo para Docentes

Acorde al rol, se diseñó el prototipo de las vistas para las funcionalidades que el docente podrá ejecutar mediante la Aplicación Móvil.

Por cada opción existe una previsualización que indica los distintos elementos que permitirán interactuar al docente referente a las Tutorías Académicas.

Menú: En la figura 27 muestra el diseño de menú para docentes.



Figura 27: Menú para docentes

Elaborado por: Paola Bautista

- **Registrar tutoría:** En la figura 28, indica el diseño de registro de tutorías por parte de los docentes.



Figura 28: Registrar tutorías por parte de los docentes

Elaborado por: Paola Bautista

Ver estados de Tutorías: En la figura 29, indica el diseño para ver el estado de las validaciones de las tutorías



Figura 29: Ver estados de las tutorías solicitadas al docente

Elaborado por: Paola Bautista

Gestión o ver estados de tutorías solicitadas: En la figura 30, indica los diseños para ver los estados de las tutorías que fueron solicitadas al docente.



Figura 30: Gestión de Tutorías Académicas

Elaborado por: Paola Bautista

3.2.3. Fases III: Producción

En la fase de producción se desarrolló la Aplicación Móvil acorde a las funcionalidades establecidas.

3.2.3.1. Backend

ASP.NET Core 5 es el framework que se usó para la creación de la Web API RESTful que permite la comunicación entre los datos y las peticiones HTTP que se realice desde el frontend desarrollado en Flutter.

Está estructurado por entidades, controladores, interfaces y servicios. Cada uno de estos componentes permite dividir responsabilidades para cumplir con las peticiones que el usuario realice desde la aplicación.

Controladores

Los controladores se encargan de ser el intermediario entre el frontend y backend para que luego se ejecute la petición HTTP necesaria para obtener el resultado deseado dependiendo de la funcionalidad en el servicio correspondiente.

Listado de Controladores:

- APINotificacionesController.cs
- ClaseController.cs
- EstadosTutDocController.cs
- EstadosTutEstController.cs
- ListadoEstudiantesController.cs
- ListadoEstValidacionesController.cs
- LoginController.cs
- MateriaDocenteController.cs
- MateriasRegistroDocController.cs
- ModificacionEstadosTutDocController.cs
- NotificacionController.cs
- NotificacionParcialController.cs
- RegistroTutoriasEstController.cs
- SolicitudController.cs
- TokenController.cs
- ValidacionTutoriasController.cs

Entidades

Las entidades son las representaciones de las tablas con sus respectivos campos.

Listado de Entidades:

- Actividade.cs
- ActividadesAcademica.cs
- APINotificacaciones.cs
- Carrera.cs

- CicloAcademico.cs
- Clase.cs
- ClaseDocDTO.cs
- ClaseDTO.cs
- DetalleTutorium.cs
- Docente.cs
- DocenteDTO.cs
- EstadosTutEstDTO.cs
- Estudiante.cs
- EstudianteDTO.cs
- Facultade.cs
- Horario.cs
- Login.cs
- Malla.cs
- Materia.cs
- MateriaDTO.cs
- MateriasDocente.cs
- MateriaTutoriaDTO.cs
- Notificacion.cs
- Solicitud.cs
- Tutoria.cs

Interfaces

Las interfaces son las encargadas de que se ejecute obligatoriamente el método necesario para cumplir con la petición HTTP.

Listado de Interfaces:

- IAPINotificaciones.cs
- IClase.cs
- IEstadosTutDoc.cs
- IEstadosTutEst.cs
- IListadoEstudiantes.cs

- IListadoEstValidaciones.cs
- IMateriaDocente.cs
- IMateriasRegistroDoc.cs
- IModificacionEstadosTutDoc.cs
- INotificacion.cs
- INotificacionParcial.cs
- IPasswordHasher.cs
- IRegistroTutoriasEst.cs
- ISolicitudService.cs
- IValidacionTutorias.cs

Servicios

Los servicios son los encargados de ejecutar la petición a la base de datos.

Listado de Servicios:

- APINotificacionesService.cs
- ClaseService.cs
- DocenteService.cs
- EstadosTutDocService.cs
- EstadosTutEstService.cs
- EstudianteService.cs
- ListadoEstudiantesService.cs
- ListadoEstValidacionesService.cs
- MateriaDocenteService.cs
- MateriasRegistroDocService.cs
- ModificacionEstadosTutDocService.cs
- NotificacionService.cs
- NotificacionParcialService.cs
- PasswordService.cs
- RegistroTutoriasEstService.cs
- SolicitudService.cs
- ValidacionTutoriasService.cs

Ejemplo de código de la Web API RESTful

La figura 31 indica el servicio ValidacionTutoriasService, para actualizar el registro de una Tutoría Académica cuando un estudiante la valida.

```
namespace WebAPI_TutoriasAcademicas.Services
{
    public class ValidacionTutoriasService : IValidacionTutorias
    {
        private readonly ApplicationDbContext _context;

        public ValidacionTutoriasService(ApplicationDbContext context)
        {
            this._context = context;
        }

        public async Task<string>
updateDetalleTutorias(List<MateriaTutoriaDTO> detalle)
        {
            foreach(MateriaTutoriaDTO vl_detalle in detalle)
            {
                var detalleBD = await
_context.DetalleTutoria.FirstOrDefaultAsync(x => x.CodDetTut ==
vl_detalle.codDetTut);
                detalleBD.VerificacionTut = vl_detalle.verificacion;
                _context.Entry(detalleBD).State = EntityState.Modified;
            }
            await _context.SaveChangesAsync();
            return "Ok";
        }
    }
}
```

Figura 31: Servicio para actualizar una tutoría cuando un estudiante la valida.

Elaborado por: Paola Bautista

La figura 32 indica la interfaz IValidacionTutorias para el uso del método updateDetalleTutorias.

```
namespace WebAPI_TutoriasAcademicas.Interfaces
{
    public interface IValidacionTutorias
    {
        Task<string> updateDetalleTutorias(List<MateriaTutoriaDTO> detalle);
    }
}
```

Figura 32: Interfaz para el uso del método updateDetalleTutorias

Elaborado por: Paola Bautista

La figura 33 indica el controlador ValidacionTutoriasController para actualizar el registro de una Tutoría Académica cuando un estudiante la valida.

```
namespace WebAPI_TutoriasAcademicas.Controllers
{
    [Authorize]
    [ApiController]
    [Route("api/[controller]")]
    public class ValidacionTutoriasController: ControllerBase
    {
        private readonly IValidacionTutorias _validacionTutorias;

        public ValidacionTutoriasController(IValidacionTutorias
validacionTutorias)
        {
            this._validacionTutorias = validacionTutorias;
        }

        [HttpPost]
        public async Task<ActionResult<string>>
updateDetalleTutorias(List<MateriaTutoriaDTO> detalle)
        {
            string resultado = await
_validacionTutorias.updateDetalleTutorias(detalle);
            return resultado;
        }
    }
}
```

Figura 33: Controlador para actualizar una tutoría cuando un estudiante la valida

Elaborado por: Paola Bautista

3.2.3.2. Storycards

El propósito de una storycard es documentar las actividades detalladamente como el tipo, la dificultad, el esfuerzo, la prioridad, y las fechas en las que se definen, implementan, ejecutan y verifican.

Storycards de Estudiantes

Las storycards para los estudiantes indican las actividades que fueron ejecutadas para cumplir con las funcionalidades acorde a los requerimientos establecidos.

Storycard - Inicio de Sesión Estudiantes

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
1	Nuevo	4	4	4	4	Alta
Descripción: Inicio de Sesión de Estudiantes						
Fecha	Estado		Comentario			
01/11/2021	Definido		Sin comentario			
08/12/2021	Implementado		Sin comentario			
13/12/2021	Ejecutado		Sin comentario			
13/12/2021	Verificado		Sin comentario			

Tabla 14: Storycard – Inicio de Sesión estudiantes.

Elaborado por: Paola Bautista

Para la funcionalidad de Inicio de Sesión para los estudiantes, se creó un método HTTP post para generar el token con tiempo de vida de 1 mes y otro método post para el inicio de sesión del estudiante implementando la seguridad por medio de un HASH con el algoritmo PBKDF2 para cifrar la contraseña. El consumo de la API como indica la figura 34.

```
Future<String> login(String email, String password) async {
  final authData = {'CorLog': email, 'PassLog': password};

  final url = Uri.http('192.168.0.107:8088', '/api/token');

  final resp = await http.post(url,
    headers: {"content-type": "application/json"},
    body: jsonEncode(authData));
  final Map<String, dynamic> usuarioResp = json.decode(resp.body);
  final tempUsuario = Usuarios.fromMap(usuarioResp);
  if (tempUsuario.rolLog != '') {
    await guardarDatos(
      tempUsuario.corLog,
      tempUsuario.token,
      tempUsuario.rolLog,
      tempUsuario.idUser,
      tempUsuario.nombre,
    );
  }
}
```



```

    obtenerNombre();
    obtenerCodigo();
    obtenerRol();
    return tempUsuario.rolLog;
} else {
    return usuarioResp['error']['message'];
}
}
}

```

Figura 34: Comunicación entre frontend y backend para el inicio de Sesión

Elaborado por: Paola Bautista

Storycard – Solicitud de Tutorías Académicas

Nº	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
2	Nuevo	4	4	4	4	Alta
Descripción: Solicitar Tutorías Académicas						
Fecha	Estado		Comentario			
01/11/2021	Definido		Sin comentario			
15/12/2021	Implementado		Sin comentario			
18/12/2021	Ejecutado		Sin comentario			
19/12/2021	Verificado		Sin comentario			

Tabla 15: Storycard - Solicitar Tutorías Académicas

Elaborado por: Paola Bautista

La funcionalidad de Solicitar Tutorías Académicas, fue desarrollada creando varios métodos HTTP con el fin de realizar peticiones de consultas sobre la materia como indica la figura 35.

```

Future<List<Clase>> getMaterias() async {
  this.isLoading = true;
  List<Clase> clase = [];
  String? usuario = await storage.read(key: 'idUser');
  String? tokenAuth = await storage.read(key: 'token');
  final url = Uri.http('192.168.0.107:8088', '/api/clase/' + usuario!);
  final resp = await http.get(url, headers: {
    "content-type": "application/json",
    'Authorization': 'Bearer $tokenAuth'
  });
  List<dynamic> claseResp = json.decode(resp.body);
  clase = List.generate(claseResp.length, (i) => Clase.from-
Map(claseResp[i]));
  obtenerCicloAcademico();
  obtenerDocente();
  obtenerMateria();
  this.isLoading = false;
  notifyListeners();
  return clase;
}

```

Figura 35: Código para obtener las materias donde el estudiante está matriculado

Elaborado por: Paola Bautista

Y los horarios disponibles del docente que imparta una tutoría de dicha materia como muestra la figura 36.

```

Future<List<Horario>> getHorarios() async {
  List<Horario> horario = [];
  this.isLoading = true;
  String? docente = '';
  String? rol = await storage.read(key: 'rol');
  if (rol == 'DOCENTE') {
    docente = await storage.read(key: 'idUser');
  } else {
    docente = await storage.read(key: 'docente');
  }
  String? tokenAuth = await storage.read(key: 'token');
  String? materia = await storage.read(key: 'materia');
  final authData = {'CodDocPer': docente, 'CodMatPer': materia};
  final url = Uri.http('192.168.0.107:8088', '/api/materiadocente');

  final resp = await http.post(url,
    headers: {
      "content-type": "application/json",
      'Authorization': 'Bearer $tokenAuth',
    },
  ),

```

```

        body: jsonEncode(authData));
final data = json.decode(resp.body);
horario = List.generate(data.length, (i) => Horario.fromMap(data[i]));
this.isLoading = false;
notifyListeners();
return horario;
}

```

Figura 36: Código para obtener horarios acordes a la materia

Elaborado por: Paola Bautista

En la figura 37, indica la ejecución de la funcionalidad para insertar el nuevo registro en la base de datos.

```

Future<String> solicitud(String codMat, String codDoc, String tema,
    int horario, List<Estudiante> est) async {
    DateTime date = DateTime.now();
    List<RegistroListTutDoc> registro = [];
    final _storage = new FlutterSecureStorage();
    String? rol = await _storage.read(key: 'rol');
    String? tokenAuth = await _storage.read(key: 'token');
    String tipoTut = '';
    String estadoTut = '';
    String fecha = DateFormat("yyyy-MM-dd").format(date).toString();
    if (rol == 'DOCENTE') {
        tipoTut = "REGISTRO";
        estadoTut = "FINALIZADA";
    } else {
        tipoTut = "SOLICITUD";
        estadoTut = "PENDIENTE";
    }
    registro.add(RegistroListTutDoc(
        codMateria: codMat,
        tema: tema,
        codDocente: codDoc,
        tipoTut: tipoTut,
        estadoTut: estadoTut,
        fechaTut: fecha,
        codHor: horario.toString(),
        verificacionTut: '',
        estudiante: est));
}

```

```

final jsons = JsonEncoder()
    .convert(registro)
    .replaceAll("\\", "")
    .replaceAll("'", "{")
    .replaceAll("}", "}");
;
final tutoria = jsons.substring(1, jsons.length - 1);

final url = Uri.http('192.168.0.107:8088', '/api/solicitud');

final resp = await http.post(url,
    headers: {
        "content-type": "application/json",
        'Authorization': 'Bearer $tokenAuth',
    },
    body: tutoria);
final String solicitudResp = resp.body;
return solicitudResp;
}

```

Figura 37: Código para insertar una solicitud de Tutoría Académica

Elaborado por: Paola Bautista

Storycard - Registro de Tutorías Académicas

Nº	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
3	Nuevo	4	4	4	4	Alta
Descripción: Registrar Tutorías Académicas						
Fecha	Estado		Comentario			
01/11/2021	Definido		Sin comentario			
20/12/2021	Implementado		Sin comentario			
26/12/2021	Ejecutado		Sin comentario			
26/01/2022	Verificado		Sin comentario			

Tabla 16: Storycard - Registro de Tutorías Académicas

Elaborado por: Paola Bautista

Para la funcionalidad de registro de tutorías se consultó las materias que el estudiante está matriculado usando el código como indica la figura 35. Acorde a la materia seleccionada se consultará la lista de tutorías que el estén pendientes por registrar como muestra figura 38.

```
Future<List<RegistroTutoriasEst>> getTutorias(
    String cedulaEst, int cicloAcademico, int materia) async {
    this.isLoading = true;
    List<RegistroTutoriasEst> tutorias = [];
    String? tokenAuth = await storage.read(key: 'token');
    final authData = {
        'cicloAcademico': cicloAcademico,
        'codEst': cedulaEst,
        'materia': materia
    };
    final url = Uri.http('192.168.0.107:8088', '/api/registorotutoriasest');

    final resp = await http.post(url,
        headers: {
            "content-type": "application/json",
            'Authorization': 'Bearer $tokenAuth',
        },
        body: jsonEncode(authData));
    List<dynamic> tutoriasResp = json.decode(resp.body);
    tutorias = List.generate(tutoriasResp.length,
        (i) => RegistroTutoriasEst.fromMap(tutoriasResp[i]));

    this.isLoading = false;
    notifyListeners();
    return tutorias;
}
```

Figura 38: Código de la consulta de las Tutorías del estudiante para validaciones

Elaborado por: Paola Bautista

Cuando el estudiante valide las tutorías, se hará el consumo del API como muestra la figura 39 para actualizar el registro con la validación

```

Future<String> updateDetalleTutorias(
    List<RegistroTutoriasEst> registro) async {
    this.isLoading = true;
    String? tokenAuth = await storage.read(key: 'token');
    final url = Uri.http('192.168.0.107:8088', '/api/validaciontutorias');

    final jsons = jsonEncode(registro.map((i) => i.toJson()).toList())
        .replaceAll("\\", "")
        .replaceAll('{', "{")
        .replaceAll('}', "}");

    final resp = await http.post(url,
        headers: {
            "Content-type": "application/json",
            'Authorization': 'Bearer $tokenAuth',
        },
        body: jsons);
    final String respuesta = resp.body;

    this.isLoading = false;
    notifyListeners();
    return respuesta;
}

```

Figura 39: Validación de tutorías

Elaborado por: Paola Bautista.

Storycard - Ver estados de las tutorías solicitadas

Nº	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
4	Nuevo	3	3	3	3	Media
Descripción: Ver estados de las tutorías solicitadas						
Fecha	Estado		Comentario			
01/11/2021	Definido		Sin comentario			
27/12/2021	Implementado		Sin comentario			
31/12/2021	Ejecutado		Sin comentario			
31/12/2021	Verificado		Sin comentario			

Tabla 17: Storycard - Ver estados de las tutorías solicitadas

Elaborado por: Paola Bautista

Para la funcionalidad de ver estados de tutorías se consultó todas las Tutorías Académicas solicitadas o registradas por el docente en el parcial actual como muestra la figura 40, con el fin de proporcionar la información del estado actual de las tutorías.

```
Future<List<EstadoTutEst>> getEstadosTutEst(String cedEst) async {
  List<EstadoTutEst> respuesta = [];
  String? tokenAuth = await storage.read(key: 'token');
  final url = Uri.http('192.168.0.107:8088', '/api/estadostutest/' +
cedEst);

  final resp = await http.post(url, headers: {
    "content-type": "application/json",
    'Authorization': 'Bearer $tokenAuth'
  });
  List<dynamic> estadosTutEstResp = json.decode(resp.body);
  respuesta = List.generate(estadosTutEstResp.length,
    (i) => EstadoTutEst.fromMap(estadosTutEstResp[i]));
  return respuesta;
}
```

Figura 40: Código para ver el estado actual de las tutorías para el estudiante

Elaborado por: Paola Bautista

Storycards Docentes

Storycard Inicio de Sesión de Docentes

Nº	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
5	Nuevo	4	4	4	4	Alta
Descripción: Inicio de Sesión de Docentes						
Fecha	Estado		Comentario			
01/11/2021	Definido		Sin comentario			
08/12/2021	Implementado		Sin comentario			
13/12/2021	Ejecutado		Sin comentario			
13/12/2021	Verificado		Sin comentario			

Tabla 18: Storycard – Inicio de Sesión de docentes

Elaborado por: Paola Bautista

Para la funcionalidad de Inicio de Sesión para los docentes, se creó un método HTTP post para generar el token con tiempo de vida de 1 mes y otro método post para el inicio de sesión del docente implementando la seguridad por medio de un HASH con el algoritmo PBKDF2 para cifrar la contraseña. El consumo de la API como indica la figura 34.

Storycard Registro de Tutorías Académicas

Nº	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
6	Nuevo	4	4	4	4	Alta
Descripción: Registro de Tutorías Académicas						
Fecha	Estado		Comentario			
01/11/2021	Definido		Sin comentario			
01/01/2022	Implementado		Sin comentario			
05/01/2022	Ejecutado		Sin comentario			
06/01/2022	Verificado		Sin comentario			

Tabla 19: Storycard - Registro de Tutorías Académicas para docentes

Elaborado por: Paola Bautista

Para la funcionalidad de Registro de Tutorías Académicas por parte de los docentes, se consultó las materias como muestra la figura 41.

```
Future<List<MateriasDocente>> getMateriasDoc() async {
  this.isLoading = true;
  List<MateriasDocente> materias = [];
  String? usuario = await storage.read(key: 'idUser');
  String? tokenAuth = await storage.read(key: 'token');
  final url =
    Uri.http('192.168.0.107:8088', '/api/materiasregistrodoc/' + usuario);

  final resp = await http.get(url, headers: {
    "content-type": "application/json",
    'Authorization': 'Bearer $tokenAuth',
  });
}
```



```

List<dynamic> tutoriasResp = json.decode(resp.body);
materias = List.generate(
    tutoriasResp.length, (i) => MateriasDocente.fromMap(tutoriasResp[i]));
obtenerCicloAcademico();
obtenerMateria();

this.isLoading = false;
notifyListeners();
return materias;
}

```

Figura 41: Código para obtener las materias que el docente imparte

Elaborado por: Paola Bautista

Y los horarios en los que el docente ha proporcionado Tutorías Académicas como se mostró en la figura 36 y para la inserción de los registros se consumió del método HTTP que indicaba la figura 37. El fin de implementar la lógica en el mismo código es para evitar que sea extenso e inentendible.

Storycard - Gestión de Estados de Tutorías Académicas

Nº	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
7	Nuevo	3	3	3	3	Media
Descripción: Gestión de Estados de Tutorías Académicas						
Fecha	Estado		Comentario			
01/11/2021	Definido		Sin comentario			
06/01/2022	Implementado		Sin comentario			
06/01/2022	Ejecutado		Sin comentario			
07/01/2022	Verificado		Sin comentario			

Tabla 20: Storycard - Gestión de Estados de Tutorías Académicas

Elaborado por: Paola Bautista

Para la funcionalidad de Tutorías Académicas se consultó las materias como muestra la figura 41. Acorde a la materia seleccionada se consultó el listado de las tutorías del parcial actual ya sea por solicitud o por registro del mismo docente como muestra la figura 42.

```

Future<List<EstadoTutEst>> getEstadosTutDoc(
    String codDoc, String materia) async {
    String? tokenAuth = await storage.read(key: 'token');
    List<EstadoTutEst> respuesta = [];
    final url = Uri.http('192.168.0.107:8088', '/api/estadostutdoc');
    final data = {"codDocTut": codDoc, "CodMatTut": materia};
    final resp = await http.post(url,
        headers: {
            "content-type": "application/json",
            'Authorization': 'Bearer $tokenAuth',
        },
        body: jsonEncode(data));
    List<dynamic> estadosTutEstResp = json.decode(resp.body);
    respuesta = List.generate(estadosTutEstResp.length,
        (i) => EstadoTutEst.fromMap(estadosTutEstResp[i]));
    return respuesta;
}

```

Figura 42: Código sobre la consulta de tutorías para docentes

Elaborado por: Paola Bautista

La actualización se realizará como muestra la figura 43.

```

Future<String> updateEstadosTutDoc(int codTut) async {
    String? tokenAuth = await storage.read(key: 'token');
    final url =
        Uri.http('192.168.0.107:8088', '/api/ModificacionEstadosTutDoc');
    final data = {"codTut": codTut};
    final resp = await http.post(url,
        headers: {
            "content-type": "application/json",
            'Authorization': 'Bearer $tokenAuth',
        },
        body: jsonEncode(data));
    final String respuesta = resp.body;

    return respuesta;
}

```

Figura 43: Actualización de estados de tutorías para docentes

Elaborado por: Paola Bautista

La actualización del estado de la tutoría se realizará si cumple con las siguientes condiciones:

- El estado como PENDIENTE.
- El horario en el que fue seleccionada la tutoría debe ser en actividades académicas debido a que el docente no puede rechazarla si es la hora asignada para la tutoría.

Storycard - Ver estados de Tutorías Académicas

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
8	Nuevo	3	3	3	3	Media
Descripción: Ver estados de Tutorías Académicas						
Fecha	Estado		Comentario			
01/11/2021	Definido		Sin comentario			
07/01/2022	Implementado		Sin comentario			
07/01/2022	Ejecutado		Sin comentario			
08/01/2022	Verificado		Sin comentario			

Tabla 21: Storycard - Ver estados de las Tutorías Académicas validadas

Elaborado por: Paola Bautista

Para la funcionalidad de ver estados de Tutorías Académicas, se consultó la materia como muestra la figura 41 y la opción que desee ver los estados de las validaciones por parte de los estudiantes. Las opciones son:

- Ver todos los estados.
- Ver las tutorías validadas.
- Ver las tutorías no validadas.
- Ver las tutorías pendientes de validar.

La lista se desplegará dependiendo de las opciones seleccionadas como muestra la figura 44.

```

Future<List<ListadoEstudiantesDoc>> obtenerListadoEstTutDoc(
    String codMat) async {
    final storage = new FlutterSecureStorage();
    String? verificacion = await storage.read(key: 'opcion');
    String? tokenAuth = await storage.read(key: 'token');
    List<ListadoEstudiantesDoc> respuesta = [];
    final url = Uri.http('192.168.0.107:8088',
        '/api/listadoestvalidaciones/' + codMat + "/" + verificacion!);

    final resp = await http.post(url, headers: {
        "content-type": "application/json",
        'Authorization': 'Bearer $tokenAuth',
    });
    List<dynamic> estadosTutEstResp = json.decode(resp.body);
    respuesta = List.generate(estadosTutEstResp.length,
        (i) => ListadoEstudiantesDoc.fromMap(estadosTutEstResp[i]));
    return respuesta;
}

```

Figura 44: Listado de tutorías acorde a las validaciones de los estudiantes

Elaborado por: Paola Bautista

Storycard – Notificaciones para docentes y estudiantes

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
9	Nuevo	3	4	4	4	Media
Descripción: Notificaciones para docentes y estudiantes						
Fecha	Estado		Comentario			
01/11/2021	Definido		Sin comentario			
09/01/2022	Implementado		Sin comentario			
16/01/2022	Ejecutado		Sin comentario			
16/01/2022	Verificado		Sin comentario			

Tabla 22: Storycard - Notificaciones para docentes y estudiantes

Elaborado por: Paola Bautista

El servicio brindado por Firebase sobre Notificaciones Push, se creó un proyecto para Android debido a que facilita la integración del plugin a la plataforma con Flutter.

Se configuró la aplicación siguiendo los pasos establecidos por la misma documentación de Firebase Cloud Messaging (FCM) en Flutter por medio de una llave para poder consumir el servicio. Se debe realizar una petición POST a su propia API para las notificaciones usando la llave de autenticación única. Además, se creó los componentes necesarios en la API RESTful creada para la aplicación como indica el código de la figura 45

```

public class APINotificacionesService: IAPINotificaciones
{
    private readonly ApplicationDbContext _context;

    public APINotificacionesService(ApplicationDbContext context)
    {
        this._context = context;
    }
    public async Task<string> notificaciones(APINotificaciones api)
    {
        using (var client = new HttpClient())
        {
            for (int i = 0; i < api.registration_ids.Count; i++)
            {
                var result = from login in _context.Logins
                             where login.IdUser == api.registration_ids[i]
                             select login;

                string token = result.ToList()[0].TokenLog;
                api.registration_ids[i] = token;
            }

            client.DefaultRequestHeaders.Authorization = new
System.Net.Http.Headers.AuthenticationHeaderValue("AAAAj9byujk:APA91bHtWEjyuhg
VC0RnZyCCNYtgHwCnYcnogreowblsSh3qqPX3Cygdmk8Vwpw-
GTHwgfjgddwXN4VxC6ReKV_wqgtY_tc0UsEquS7-m5BujL4axh60yIteY12XjleTg51UzQSHTY");
            StringContent content = new
StringContent(JsonConvert.SerializeObject(api), Encoding.UTF8,
"application/json");

            using (var response = await
client.PostAsync("https://fcm.googleapis.com/fcm/send", content))
            {
                string apiresponse = await
response.Content.ReadAsStringAsync();
            }

        }
        return "Ok";
    }
}

```

Figura 45: Consumo de la API Firebase Cloud Messaging

Elaborado por: Paola Bautista

El código de las notificaciones fue implementado en las clases que ejecutan las funcionalidades determinadas para cada petición solicitada consumiendo el servicio APINotificacionesService.

En el caso de las notificaciones que se deben ejecutar en un horario específico. Se creó en otras clases usando un paquete propio de Visual Studio llamado HostedService que permite realizar estas acciones.

Notificación para el docente que tiene una nueva solicitud de tutoría o para el estudiante que tiene una tutoría pendiente de validar.

Para esta actividad se implementó código en el servicio de solicitud para notificar al docente que tiene una nueva solicitud de tutoría como muestra la figura 46.

```
if (tutoria.TipoTut == "SOLICITUD")
{
    List<string> docentes = new List<string>();
    docentes.Add(tutoria.CodDocTut);
    vl_data.tema = tutoria.TemaTut;
    vl_data.tipo = tutoria.TipoTut;
    vl_data.pantalla = "estadosDoc";
    notify.data = vl_data;
    vl_notification.body = "Tiene una solicitud sobre el tema
" + tutoria.TemaTut + ". Hora de tutoría: " + horario.ToList()[0].HoraFin;
    vl_notification.title = "SOLICITUD DE TUTORÍA ACADÉMICA";
    notify.notification = vl_notification;
    notify.priority = "high";
    notify.registration_ids = docentes;

    await aPINotificaciones.notificaciones(notify);
    TimeSpan time1 = TimeSpan.FromHours(1);
    Notificacion notificacion2 = new Notificacion
    {
        cod_tut = id,
        hora_not = horario.ToList()[0].HoraFin.Add(time1),
        cuerpo_not = "Tiene una nueva tutoría pendiente por
validar sobre el tema: " + tutoria.TemaTut,
        titulo_not = "VALIDACIÓN DE TUTORÍA ACADÉMICA",
        pantalla = "registroEst",
        tipo_not = "VALIDACION"
    };
    await _context.Notificaciones.AddAsync(notificacion2);
}
```

Figura 46: Código - Notificación para el docente que tiene una nueva solicitud de tutoría

Elaborado por: Paola Bautista

En la figura 47, indica el código para notificar al estudiante que una nueva tutoría pendiente para validar.

```
else if (tutoria.TipoTut == "REGISTRO")
{
    List<string> idEst = new List<string>();
    foreach (Estudiante item in solicitud.estudiante)
    {
        idEst.Add(item.CodEst);
    }
    vl_data.tema = tutoria.TemaTut;
    vl_data.tipo = tutoria.TipoTut;
    vl_data.pantalla = "estadosDoc";
    notify.data = vl_data;
    vl_notification.body = "Tiene una nueva tutoría pendiente
por validar sobre el tema: " + tutoria.TemaTut;
    vl_notification.title = "VALIDACIÓN DE TUTORÍA ACADÉMICA";
    notify.notification = vl_notification;

    notify.priority = "high";
    notify.registration_ids = idEst;
    await aPINotificaciones.notificaciones(notify);
}

await _context.SaveChangesAsync();
tutoria = null;

return "correcto";
}
```

Figura 47: Código - Notificación para el estudiante que tiene una tutoría pendiente de validar

Elaborado por: Paola Bautista

Notificación para los estudiantes cuando el docente rechazó una tutoría

El código fue implementado en la clase que se encarga de ejecutar la actualización de la tutoría a estado “RECHAZADA” siempre y cuando cumpla con las condiciones establecidas para esta funcionalidad. Luego de la actualización, se ejecuta el código para notificar al estudiante que su petición fue rechazada, como indica la figura 48.

```

var detalleTutoria = from detalle in _context.DetalleTutoria
                    where detalle.CodTut == tut.CodTut
                    select detalle;

var tuto = from tutorias in _context.Tutorias
           where tutorias.CodTut == tut.CodTut
           select tutorias;

var docente = from doc in _context.Docentes
              where doc.CodDoc == tuto.ToList()[0].CodDocTut
              select doc;

codEst = detalleTutoria.ToList()[0].CodEstTut;

List<string> estudiantes = new List<string>();
estudiantes.Add(codEst);
tema = tuto.ToList()[0].TemaTut;
nombreDoc = docente.ToList()[0].Nom1Doc +
docente.ToList()[0].Ape1Doc;
APINotificaciones notify = new APINotificaciones();
Data vl_data = new Data();
Notification vl_notification = new Notification();
vl_data.tema = tema;
vl_data.tipo = "SOLICITUD RECHAZADA";
vl_data.pantalla = "estadostutest";
notify.data = vl_data;
vl_notification.body = "El docente " + nombreDoc + " rechazó la
solicitud de la tutoría: " + tema;
vl_notification.title = "SOLICITUD RECHAZADA";
notify.notification = vl_notification;
notify.priority = "high";
notify.registration_ids = estudiantes;

await aPINotificaciones.notificaciones(notify);

```

Figura 48: Código - Notificación en caso de que se rechace una solicitud de tutoría de un estudiante

Elaborado por: Paola Bautista

Notificaciones programadas

El siguiente código indica la clase NotificacionService. Esta clase es la encargada de realizar toda la lógica para notificar a los estudiantes después de una hora que deben validar la tutoría que habían solicitado como indica la figura 49.


```

        public NotificacionService(ApplicationDbContext context,
IAPINotificaciones aPINotificaciones)
        {
            this._context = context;
            this.aPINotificaciones = aPINotificaciones;
        }

        private readonly ApplicationDbContext _context;
        private readonly IAPINotificaciones aPINotificaciones;

        public NotificacionService(ApplicationDbContext context,
IAPINotificaciones aPINotificaciones)
        {
            this._context = context;
            this.aPINotificaciones = aPINotificaciones;
        }
        public async Task<string> procesoNotificaciones()
        {
            TimeSpan myDateResult = DateTime.Now.TimeOfDay;
            List<Notificacion> notificacion;
            notificacion = await (from not in _context.Notificaciones
                                join cl in _context.Tutorias on not.cod_tut
                                equals cl.CodTut into noti
                                from m in noti.DefaultIfEmpty()
                                where m.EstadoTut == "PENDIENTE" &&
                                not.hora_not <= myDateResult && m.TipoTut == "SOLICITUD"
                                select new Notificacion()
                                {
                                    cod_not = not.cod_not,
                                    hora_not = not.hora_not,
                                    cod_tut = not.cod_tut,
                                    cuerpo_not = not.cuerpo_not,
                                    pantalla = not.pantalla,
                                    tipo_not = not.tipo_not,
                                    titulo_not = not.titulo_not
                                }).ToListAsync();

            APINotificaciones notify = new APINotificaciones();
            Data vl_data = new Data();
            Notification vl_notification = new Notification();
            string codEst = "";
            string nombreDoc = "";
            foreach (Notificacion n in notificacion)
            {
                Tutoria tut = await _context.Tutorias.FirstOrDefaultAsync(x =>
x.CodTut == n.cod_tut);

                var docente = from doc in _context.Docentes
                               where doc.CodDoc == tut.CodDocTut
                               select doc;
                var detalleTutoria = from detalle in _context.DetalleTutoria
                                     where detalle.CodTut == tut.CodTut
                                     select detalle;
            }
        }
    }

```

```

        codEst = detalleTutoria.ToList()[0].CodEstTut;
        nombreDoc = docente.ToList()[0].Nom1Doc + " " +
docente.ToList()[0].Ape1Doc;
        List<string> estudiantes = new List<string>();
        estudiantes.Add(codEst);
        vl_data.tema = tut.TemaTut;
        vl_data.tipo = "VALIDACIÓN";
        vl_data.pantalla = "registroEst";
        notify.data = vl_data;
        vl_notification.body = "Tiene una nueva tutoría pendiente por
validar sobre el tema: " + tut.TemaTut;
        vl_notification.title = "VALIDACIÓN DE TUTORÍA";
        notify.notification = vl_notification;
        notify.priority = "high";
        notify.registration_ids = estudiantes;
        await aPINotificaciones.notificaciones(notify);
        tut.EstadoTut = "FINALIZADA";
        _context.Entry(tut).State = EntityState.Modified;
        var borrarNotificacion = await
_context.Notificaciones.FirstOrDefaultAsync(x => x.cod_tut == n.cod_tut);
        _context.Notificaciones.Remove(borrarNotificacion);

    }
    await _context.SaveChangesAsync();

    return "Ok";
}
}
}

```

Figura 49: Código - Notificación programada para que el estudiante valide una tutoría después de una hora.

Elaborado por: Paola Bautista

Para las notificaciones al finalizar cada parcial, se implementó la clase NotificacionParcialService que es la encargada de realizar la lógica de la funcionalidad como indica la figura 50.

```

public class NotificacionParcialService : INotificacionParcial
{
    private readonly ApplicationDbContext _context;
    private readonly IAPINotificaciones aPINotificaciones;
    public NotificacionParcialService(ApplicationDbContext context,
IAPINotificaciones aPINotificaciones)
    {
        this._context = context;
        this.aPINotificaciones = aPINotificaciones;
    }
    public async Task<string> notificacionesParcial()
    {
        DateTime fechaFinParcial;
        DateTime fechaIniParcial;
        DateTime fechaActual = DateTime.Now;
        DateTime fechaAux;
    }
}

```

```

        List<ActividadesAcademica> parcial = await (from act in
_context.ActividadesAcademicas
                                join cic in
_context.CicloAcademicos on act.CodCicPer equals cic.CodCic into ciclo
                                from m in
ciclo.DefaultIfEmpty()
                                where m.EstCic == 1 &&
(fechaActual >= act.FechaIni && fechaActual <= act.FechaFin)
                                && (act.CodActAcad ==
1 || act.CodActAcad == 2)
                                select new
ActividadesAcademica()
                                {
                                    FechaFin =
act.FechaFin,
                                    FechaIni =
act.FechaIni
                                }).ToListAsync();

        fechaFinParcial = parcial[0].FechaFin;
        fechaAux = fechaFinParcial;
        fechaIniParcial = parcial[0].FechaIni;
        fechaFinParcial = fechaFinParcial.AddDays(-7);
        if (fechaActual >= fechaFinParcial && fechaActual <= fechaAux)
        {
            List<string> idsEstudiantes = new List<string>();
            var detalleTutorias = _context.DetalleTutoria.Where(x =>
x.VerificacionTut == "" && x.FechaTut >= fechaIniParcial && x.FechaTut <=
fechaAux).ToListAsync();
            foreach (DetalleTutorium detalle in await
detalleTutorias.ConfigureAwait(false))
            {
                string idEst = detalle.CodEstTut;
                bool existe = idsEstudiantes.ToList().Any(x => x ==
idEst);

                if (!existe)
                {
                    idsEstudiantes.Add(detalle.CodEstTut);
                }
            }
            APINotificaciones notify = new APINotificaciones();
            Data vl_data = new Data();
            Notification vl_notification = new Notification();
            vl_data.tema = "Validaciones";
            vl_data.tipo = "VALIDACIÓN";
            vl_data.pantalla = "registroEst";
            notify.data = vl_data;
            vl_notification.body = "El parcial está finalizando y le
recordamos que tiene tutorías pendientes por validar.";
            vl_notification.title = "VALIDACIÓN DE TUTORÍAS";
            notify.notification = vl_notification;
            notify.priority = "high";

```

Figura 50: Código - Notificación programada una semana antes de finalizar cada parcial

Elaborado por: Paola Bautista

Finalmente, se llama los métodos en el HostedService para programar las notificaciones acordes a su funcionalidad. Para poder llamar a los métodos, se inyectó los mismos por medio de las interfaces como indica la figura 51.

```

private readonly INotificacion notificacion;
private readonly INotificacionParcial notificacionParcial;
private readonly IServiceScopeFactory factory;
private Timer _timer;
private Timer _timer2;
public IntervalTaskHostedService(IServiceScopeFactory factory)
{
    this.factory = factory;
    notificacion =
factory.CreateScope().ServiceProvider.GetService<INotificacion>();
    notificacionParcial =
factory.CreateScope().ServiceProvider.GetService<INotificacionParcial>();
}

public Task StartAsync(CancellationTokens cancellationTokens)
{
    _timer = new Timer(notificarValidacion,
        null,
        TimeSpan.Zero,
        TimeSpan.FromHours(1));
    _timer2 = new Timer(notificacionFinParcial,
        null,
        TimeSpan.Zero,
        TimeSpan.FromDays(30));
    return Task.CompletedTask;
}

public async void notificarValidacion(object state)
{
    await notificacion.procesoNotificaciones();
}
public async void notificacionFinParcial(object state)
{
    await notificacionParcial.notificacionesParcial();
}

public Task StopAsync(CancellationTokens cancellationTokens)
{
    _timer?.Change(Timeout.Infinite, 0);
    _timer2?.Change(Timeout.Infinite, 0);
    return Task.CompletedTask;
}

public void Dispose()
{
    _timer?.Dispose();
    _timer2?.Dispose();
}

```

Figura 51: HostedService para tareas programadas

Elaborado por: Paola Bautista

3.2.4. Fase IV: Estabilización

La fase de estabilización se centra en los cambios realizados que fueron necesarios para el desarrollo de la Aplicación Móvil.

Referente al diseño y modelamiento de la base de datos se realizó cambios debido a que era necesario agregar y modificar tablas.

Los cambios fueron:

- Agregación y relacionamiento de la tabla CLASES y con ESTUDIANTES, DOCENTES, HORARIO y CICLO_ACADÉMICO.
- Agregación de los campos cod_hor (Horario), tip_mat_hor (Tipo de horario) en la tabla MATERIAS_DOCENTES.

Tipo de horario:

- Tutoría – T.
- Clase – C.
- Actividad académica – A.
- Actividad PAE – AP.
- Se eliminó la tabla MATERIAS_ESTUDIANTES.
- Se agregó la tabla auxiliar NOTIFICACIONES.

3.2.5. Fase V: Pruebas

Acorde a la fase cinco de la Metodología Mobile – D, se realizó las pruebas correspondientes que permitió verificar y evaluar el correcto funcionamiento de las funcionalidades desarrolladas con el fin de corregir y mejorar las mismas.

3.2.5.1. Pruebas de las funcionalidades

Login de docentes y estudiantes

Para el login de docentes y estudiantes se debe digitar el correo y contraseña institucional como indica la figura 52.



Figura 52: Inicio de sesión para docentes y estudiantes

Elaborado por: Paola Bautista

Menú de estudiantes y estudiantes

En el menú se muestran las funcionalidades que el estudiante y el docente puede ejecutar acorde a su rol como muestra la figura 53.

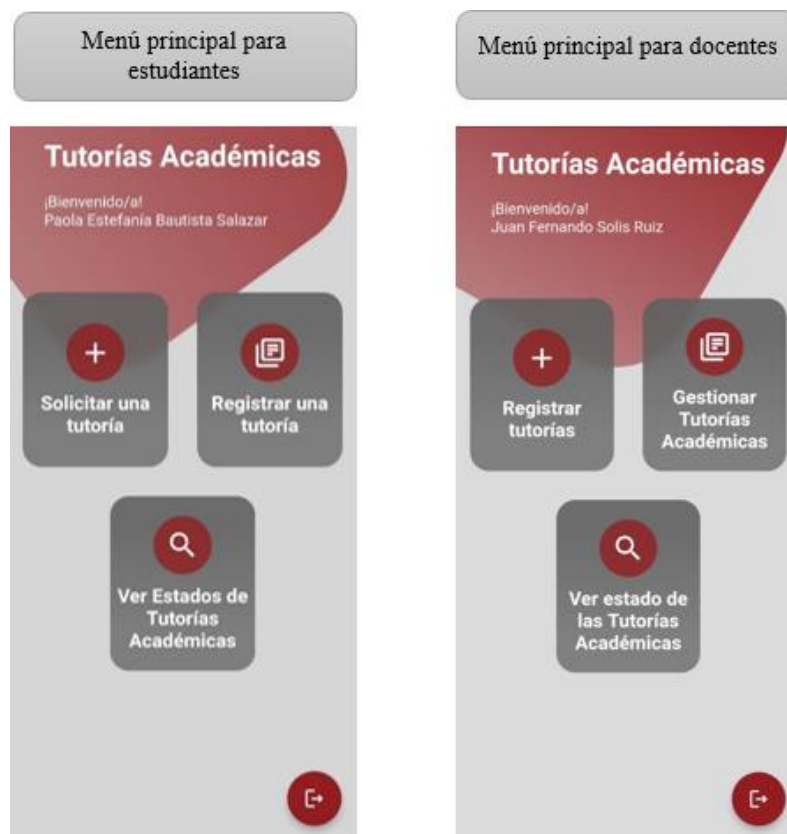


Figura 53: Menú para estudiantes y docentes respectivamente

Elaborado por: Paola Bautista

Funcionalidades para estudiantes

Acorde a las funcionalidades establecidas para los estudiantes, las siguientes figuras muestran los resultados provenientes de las pruebas realizadas en la Aplicación Móvil.

Solicitud de Tutorías

El estudiante digita el tema, selecciona la materia y el horario disponible para la tutoría. Las materias que puede seleccionar son en las que el estudiante se encuentra matriculado y el horario aparecerá dependiendo de la materia y si en ese día aún tiene horas disponibles considerando las horas de actividades académicas, acorde a lo establecido para las funcionalidades. La figura 54 muestra el resultado de la prueba ejecutada.

← Solicitar tutoría

Tema

Materia Seleccione un... >

Horario Seleccione un ... >

Solicitar

Figura 54: Solicitud de tutorías

Elaborado por: Paola Bautista

Validación de Tutorías

El estudiante selecciona la materia y al dar en el botón de siguiente le redireccionará a la ventana de listado de tutorías por validar de dicha materia.

El resultado obtenido cumple al 100% con la funcionalidad como indica la figura 55. Por lo tanto, el requerimiento establecido referente a las validaciones de tutorías por parte de los estudiantes.

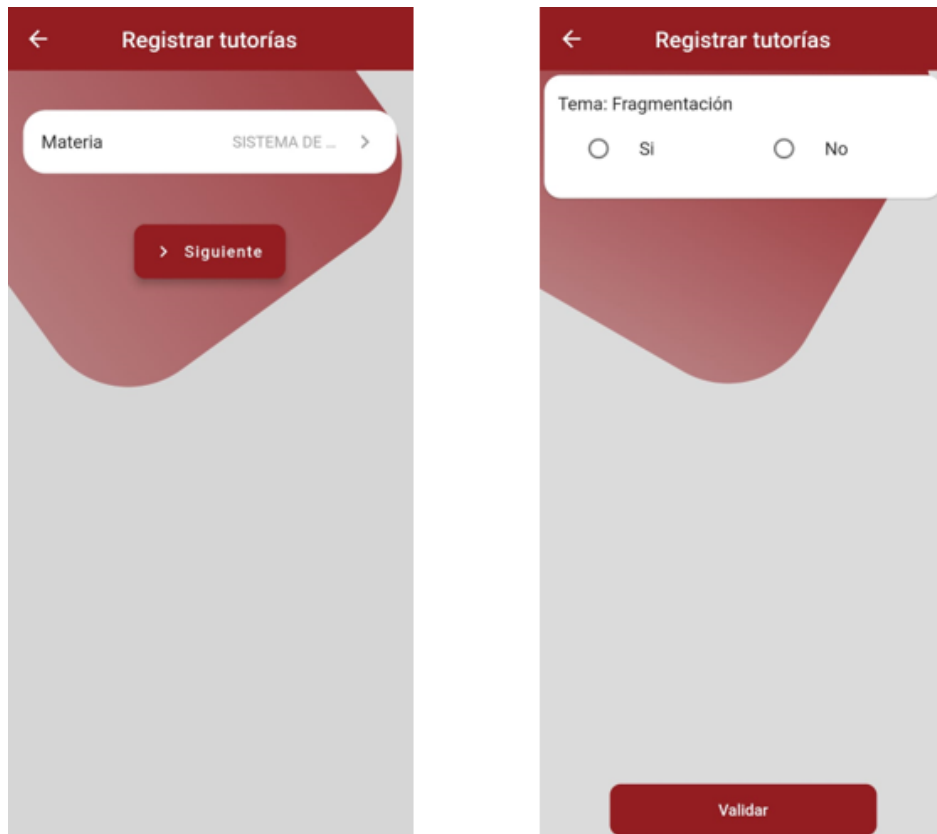


Figura 55: Validación de Tutorías Académicas

Elaborado por: Paola Bautista

Ver estados de las tutorías Académicas

Los datos que podrá visualizar son:

- Tema de la tutoría.
- Materia solicitada o registrada.
- Fecha de solicitud o de registro.
- Docente que imparte la materia.
- Horario de la tutoría.
- Estado de la tutoría.

El estudiante puede ver de manera detallada las Tutorías Académicas y sus estados actuales como indica la figura 56.

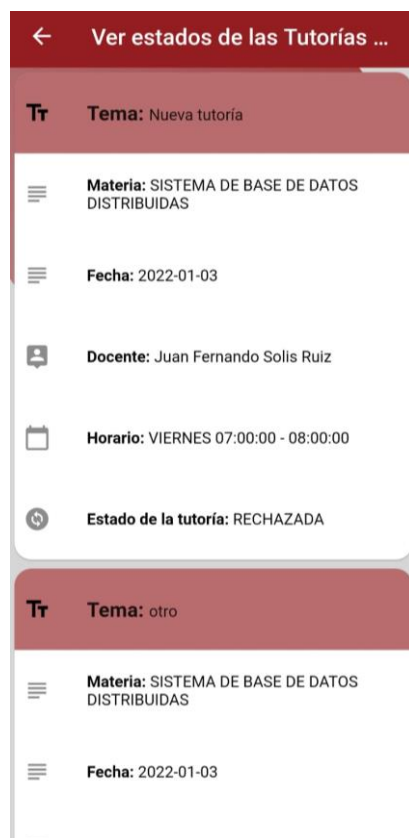


Figura 56: Estados de las tutorías para estudiantes

Elaborado por: Paola Bautista

Funcionalidades para los Docentes

Acorde a las funcionalidades establecidas para docentes, las siguientes figuras mostrarán los resultados.

Registro de Tutorías Académicas

El docente puede registrar las Tutorías Académicas digitando el tema, seleccionando la materia y el horario que impartió la tutoría, dando click en siguiente, este le redireccionará al listado de estudiantes que estén matriculados en la materia que previamente ha seleccionado. El estado de la tutoría se guardará como FINALIZADA.

La figura 57 muestra el resultado de la funcionalidad.

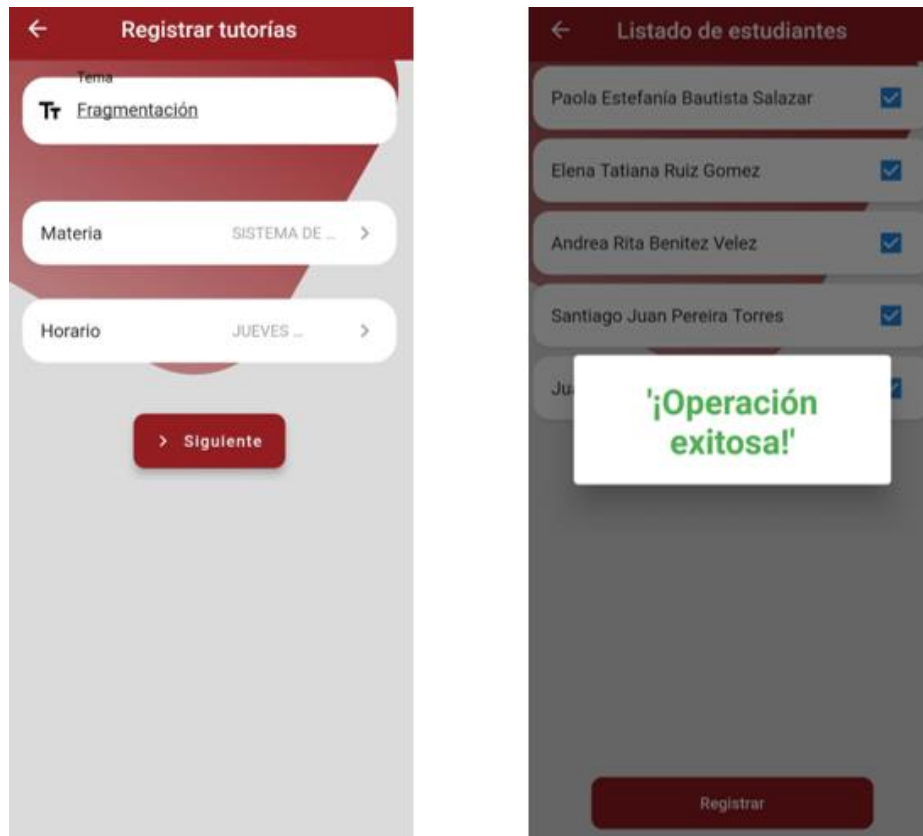


Figura 57: Registro de Tutorías Académicas para los docentes

Elaborado por: Paola Bautista

Gestión de Tutorías Académicas

La gestión de Tutorías Académicas permite modificar el estado de la tutoría siempre y cuando cumpla con las condiciones de que el estado de la tutoría debe ser PENDIENTE y que el horario en el que fue solicitada sea en horas de actividades académicas. Esto es debido a que acorde al reglamento de las Tutorías Académicas de la Universidad Técnica de Ambato, no puede rechazar una petición de un estudiante sobre un tema de la materia que imparte el docente en el horario asignado, sin embargo, en el caso de que el estudiante solicite una tutoría en la hora de actividad académica de la materia, no está en la obligación de tutorar.

La decisión es del docente si desea o no ayudarle al estudiante. La figura 58 muestra el resultado de la funcionalidad.

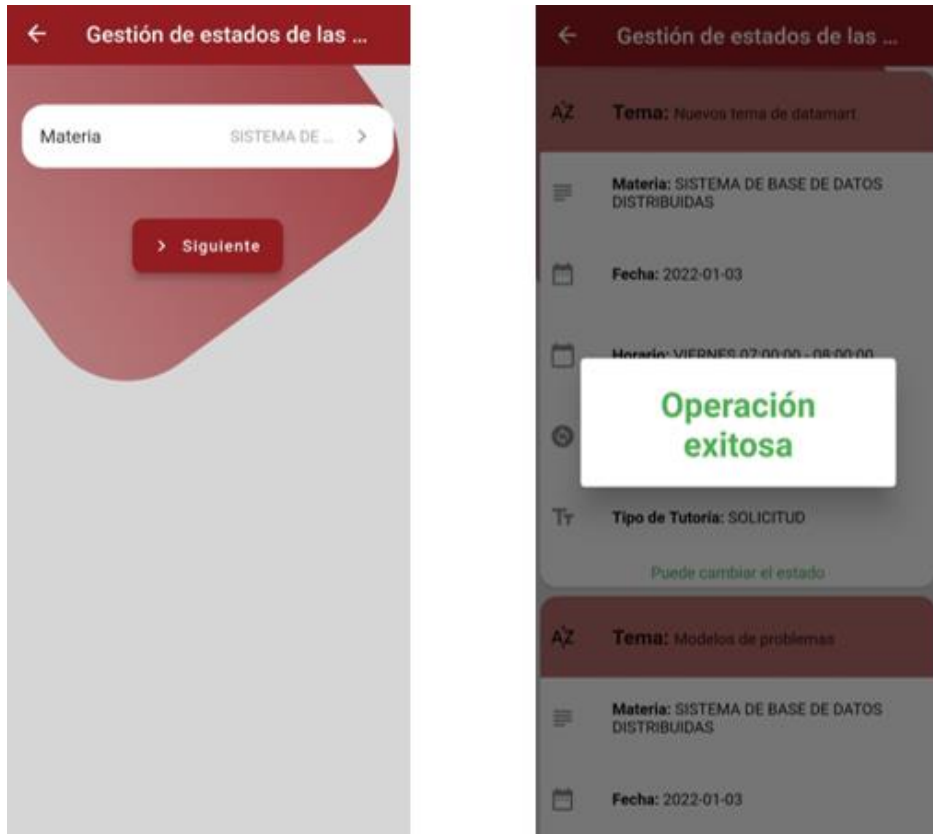


Figura 58: Gestión de Tutorías Académicas

Elaborado por: Paola Bautista

Ver estados de las Tutorías Académicas

Los docentes podrán ver el estado de las Tutorías Académicas seleccionando la materia y las opciones.

Las opciones son:

- Ver todos.
- Ver las tutorías validadas.
- Ver las tutorías no validadas.
- Ver las tutorías pendientes de validar.

La figura 59 muestra el resultado de la ejecución de la funcionalidad.

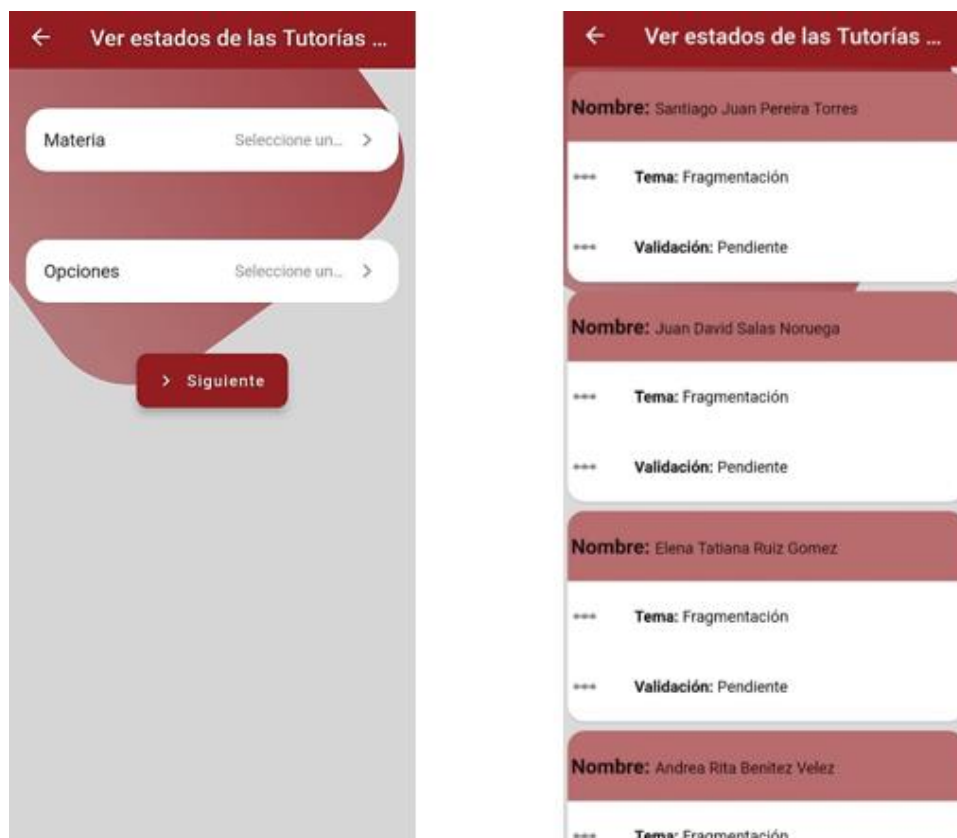


Figura 59: Ver los estados de las Tutorías Académicas para docentes

Elaborado por: Paola Bautista

Notificaciones

- **Notificación para docente**

El docente recibirá una notificación cuando un estudiante solicite una tutoría como muestra la figura 60.

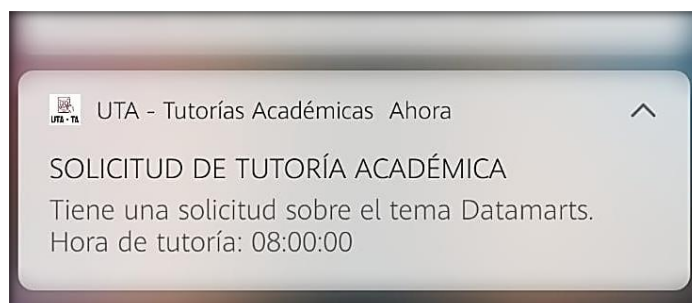


Figura 60: Resultado de notificación cuando recibe una solicitud

Elaborado por: Paola Bautista

Notificaciones para estudiantes

- **Notificación para validar después de una hora las tutorías solicitadas**

El estudiante recibirá la notificación como indica la figura 61, una hora después de haber solicitado una tutoría.

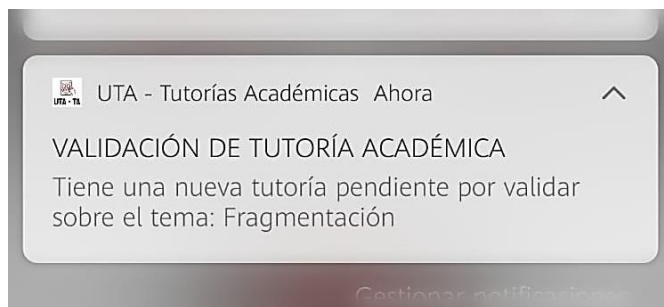


Figura 61: Resultado de notificación una hora después de haber solicitado una tutoría

Elaborado por: Paola Bautista

- **Notificación cuando el docente rechace una solicitud**

El estudiante recibirá una notificación de Solicitud Rechazada cuando el docente cambie el estado de una solicitud como indica la figura 62.

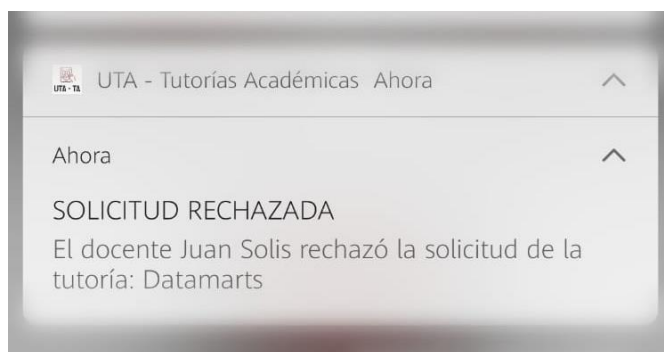


Figura 62: Notificación de solicitud rechazada

Elaborado por: Paola Bautista

- **Notificación antes de una semana de finalizar el parcial**

El estudiante recibirá una notificación como indica la figura 63 una semana antes de que se termine el parcial con el fin de recordarle en caso de que tenga tutorías pendientes por validar.



Figura 63: Resultado de notificación una semana antes de finalizar el parcial

Elaborado por: Paola Bautista

3.2.5.2. Pruebas de aceptación

Acorde a las pruebas realizadas sobre las funcionalidades de la Aplicación Móvil, se obtuvo las siguientes pruebas de aceptación.

Prueba de aceptación	Número: 1
Nº de Storycard: 1	
Nombre: Inicio de Sesión para docentes y estudiantes	
Descripción: La funcionalidad permite la autenticación del usuario para el ingreso a las demás funcionalidades.	
Condiciones de ejecución e interfaz: Se visualiza la pantalla con los campos de entrada de correo institucional y contraseña. El botón permite el inicio de sesión si los datos son correctos.	
Resultado esperado: Si los datos coinciden, la aplicación se encargará de enviar al menú principal dependiendo del rol, esto quiere decir que, si es estudiante o docente, se mostrará en el menú las funcionalidades acordes al rol.	
Resultado de la prueba: Prueba satisfactoria.	

Tabla 23: Prueba de aceptación 1

Elaborado por: Paola Bautista

Prueba de aceptación	Número: 2
Nº de Storycard: 2	
Nombre: Solicitar Tutorías Académicas	
Descripción: La funcionalidad permite al estudiante solicitar tutorías a los docentes que impartan la materia donde tengas dudas.	
Condiciones de ejecución e interfaz: El estudiante debe digitar el tema, seleccionar la materia y luego el horario disponible y el botón de solicitar se habilitará si los datos son los correctos.	
Resultado esperado: Se registrará la solicitud y la tutoría y también se enviará una notificación 15 minutos antes (del horario seleccionado por el estudiante) al docente tutor que tiene una solicitud.	
Resultado de la prueba: Prueba satisfactoria.	

Tabla 24: Prueba de aceptación 2

Elaborado por: Paola Bautista

Prueba de aceptación	Número: 3
Nº de Storycard: 3	
Nombre: Validar Tutorías Académicas	
Descripción: La funcionalidad permite al estudiante validar las tutorías a las que asistió.	
Condiciones de ejecución e interfaz: El estudiante debe seleccionar la materia que desee validar las tutorías y dar click en el botón de siguiente, este le enviará a la lista de tutorías asistidas pendientes por validar, le mostrará dos opciones de SI o NO.	
Resultado esperado: Se guardará la validación acorde a la opción seleccionada por el estudiante.	
Resultado de la prueba: Prueba satisfactoria.	

Tabla 25: Prueba de aceptación 3

Elaborado por: Paola Bautista

Prueba de aceptación	Número: 4
Nº de Storycard: 4	
Nombre: Ver estados de las Tutorías Académicas para los estudiantes	
Descripción: La funcionalidad permite al estudiante ver los estados de las tutorías con fines informativos.	
Condiciones de ejecución e interfaz: El estudiante debe seleccionar la materia y se desplegará el listado de las tutorías	
Resultado esperado: Se mostrará un listado de las tutorías del parcial actual y la información será: el tema, fecha de tutoría, docente tutor, estado, horario.	
Resultado de la prueba: Prueba satisfactoria.	

Tabla 26: Prueba de aceptación 4

Elaborado por: Paola Bautista

Prueba de aceptación	Número: 5
Nº de Storycard: 5	
Nombre: Registro de Tutorías Académicas	
Descripción: La funcionalidad permite al docente registrar una tutoría en caso de que los estudiantes no lo hayan solicitado por medio de la aplicación.	
Condiciones de ejecución e interfaz: El docente debe digitar el tema, seleccionar la materia y horario en el que impartió una tutoría, luego da en el botón de siguiente y le aparecerá una lista de los estudiantes que estén matriculados en la materia y seleccionado a los que asistieron a dicha tutoría.	
Resultado esperado: Se guardará en registro de la tutoría y se enviará una notificación a los estudiantes seleccionados por el docente tutor.	
Resultado de la prueba: Prueba satisfactoria.	

Tabla 27: Prueba de aceptación 5

Elaborado por: Paola Bautista

Prueba de aceptación	Número: 6
Nº de Storycard: 6	
Nombre: Gestión de estados de las Tutorías Académicas	
Descripción: La funcionalidad permite al docente modificar el estado de una tutoría	
Condiciones de ejecución e interfaz: El docente debe seleccionar la materia que desee modificar su estado. Luego le mostrará la lista de las tutorías de dicha materia y le aparecerá el mensaje de que puede modificar su estado siempre y cuando sea la tutoría solicitada y el horario sea en hora de actividades académicas. Esta funcionalidad solo permite rechazar solicitudes si cumplen con las condiciones mencionadas.	
Resultado esperado: Se modificará el estado de la tutoría a RECHAZADA y se le notificará al estudiante sobre el mismo.	
Resultado de la prueba: Prueba satisfactoria.	

Tabla 28: Prueba de aceptación 6

Elaborado por: Paola Bautista

Prueba de aceptación	Número: 7
Nº de Storycard: 7	
Nombre: Ver estados de las Tutorías Académicas	
Descripción: La funcionalidad permite al docente ver los estados de las tutorías.	
Condiciones de ejecución e interfaz: El docente debe seleccionar la materia que desee tener información sobre las tutorías y también debe seleccionar la opción de Ver todos, Ver los validados, Ver los no validados y Ver las pendientes por validar.	
Resultado esperado: Al docente se le mostrará la lista de tutorías y la siguiente información: El estudiante, tema, fecha, horario, y estado.	
Resultado de la prueba: Prueba satisfactoria.	

Tabla 29: Prueba de aceptación 7

Elaborado por: Paola Bautista

Prueba de aceptación	Número: 8
Nº de Storycard: 8	
Nombre: Notificaciones	
Descripción: La funcionalidad permite tanto al docente y al estudiante recibir notificaciones	
Condiciones de ejecución e interfaz: El docente o estudiante recibirá una notificación dependiendo de las funcionalidades ejecutadas.	
Resultado esperado: <ul style="list-style-type: none"> • Si el estudiante solicitó una tutoría, le llegará al docente una notificación de la tutoría 15 minutos antes de la hora de inicio. • Si el docente rechaza una solicitud, le llegará al estudiante una notificación de la solicitud rechazada. • Si el docente registra una tutoría, le llegará la notificación al/los estudiantes que fueron seleccionados para que puedan validar la tutoría. • Después de haberse dado a lugar la tutoría solicitada por el estudiante, le llegará una notificación después de finalizar la misma para que pueda validarla. • Una semana antes de finalizar el parcial, les notificará a todos los estudiantes que tengan tutorías pendientes por validar para que efectúen dicha validación. 	
Resultado de la prueba: Prueba satisfactoria.	

Tabla 30: Prueba de aceptación 8

Elaborado por: Paola Bautista

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

A continuación, se listan las conclusiones referentes al presente proyecto.

- El análisis de datos procedentes de los resultados obtenidos de la entrevista y encuestas aplicadas a los stakeholders, permitieron establecer de manera correcta los requerimientos y funcionalidades de la Aplicación Móvil para las Tutorías Académicas.
- La investigación realizada sobre el Framework Flutter, facilitó tener una comprensión más clara sobre el uso del mismo, haciendo posible el desarrollo y brindando una experiencia agradable, a pesar de ser un Framework reciente, sin embargo, es lo suficientemente robusto, ofrece variedad de complementos y componentes necesarios para que el resultado de la Aplicación Móvil planteada cumpla con lo requerido.
- La metodología ágil utilizada Mobile – D permitió que el desarrollo del presente proyecto sea rápido y fácil de ejecutar, debido a que la metodología fue diseñada para ser aplicada en proyectos pequeños y en tiempos cortos de entrega.
- La Aplicación Móvil cumple con las funcionalidades establecidas acorde a los requerimientos, ayuda a mejorar la interacción entre los docentes y estudiantes con las Tutorías Académicas, convirtiéndose en una herramienta útil para las actividades del día a día de los interesados del presente proyecto.

4.2. Recomendaciones

- Se recomienda capacitar a los encargados del mantenimiento de la Aplicación Móvil.
- Es recomendable fomentar el uso de nuevas tecnologías entre estudiantes y docentes, tales como el uso de teléfonos inteligentes, ya que esto permitirá aumentar el interés y facilitará la interacción con las Tutorías Académicas.
- Sería de utilidad profundizar la investigación sobre el Framework Flutter con el fin de llevar mejoras continuas a la Aplicación Móvil.
- Se debe considerar tener un correcto mantenimiento a la base de datos que se maneja para las Tutorías Académicas con el fin de evitar inconvenientes que se puedan presentar durante el uso de la Aplicación Móvil.

BIBLIOGRAFÍA

- [1] L. A. Perú, “Universidad para el desarrollo andino,” pp. 1–174, 2015.
- [2] F. Vargas, “Sistema Integrado de Autenticación para la Universidad Tecnológica de Bolívar- MiUTB,” 2016, [Online]. Available: <https://repositorio.utb.edu.co/bitstream/handle/20.500.12585/486/0069824.pdf?sequence=1>.
- [3] S. Ronquillo, “Sistema integrado de auditoría educativa para mejorar el proceso de evaluación docente en las universidades estatales de la Zona Ocho del Ecuador,” 2019.
- [4] T. N. Vaca, Modelo De Calidad De Software Aplicado Al Módulo De Talento Humano Del Sistema Informático Integrado Universitario – UTN. 2017.
- [5] I. Sommerville, Ingeniería del Software, Séptima Ed. España: Person Education, 2006.
- [6] L. Herazo, “¿Qué Es Una Aplicación Móvil?,” Tecnología, p. 1, 2020, [Online]. Available: <https://anincubator.com/que-es-una-aplicacion-movil/>.
- [7] flutter.dev, “Flutter - Beautiful native apps in record time,” 2021. https://flutter.dev/?gclid=Cj0KCQjwzYGGBhCTARIsAHdMTQzARP7zCijzkwlixDnzksdxbs8_DBpi3xM17_axRnF9lyq_Y5rNFmwaAvsbEALw_wcB&gclidsrc=aw.ds.
- [8] F. Gavilán, “Udemy,” 2021. <https://www.udemy.com/course/construyendo-web-apis-restful-con-aspnet-core>.
- [9] Mozilla, “Developer Mozilla,” 2021. <https://developer.mozilla.org/es/docs/Web/HTTP/Status>.
- [10] Consejo de Educación Superior, “Reglamento de Régimen Académico de las Universidades del Ecuador.”
- [11] uta.edu.ec, Reglamento de las Tutorías Académicas y Grupales de la Universidad Técnica de Ambato. 2017.
- [12] N. Chicaiza and N. Guañotuña, “Sistema Integrado para la gestión de

indicadores de calidad de la Facultad de CIYA: Proyectos/ Programas de Vinculación con la Sociedad,” vol. 1, p. 101, 2018, [Online]. Available: <http://repositorio.utc.edu.ec/bitstream/27000/4501/1/PI-000727.pdf>.

- [13] “Instructivo del Sistema Integrado de Información de la Universidad Técnica de Ambato ‘SI-UTA,’” 2019.

ANEXOS

ANEXO A

Manual de usuario

"APLICACIÓN MÓVIL USANDO EL FRAMEWORK FLUTTER PARA EL CONTROL DE LAS TUTORÍAS ACADÉMICAS DEL SISTEMA INTEGRADO DE LA UNIVERSIDAD TÉCNICA DE AMBATO"

MANUAL DE USUARIO

El presente manual de usuario fue desarrollado con el fin de facilitar y guiar al docente y estudiante con las diferentes funcionalidades de la Aplicación Móvil desarrollada sobre las Tutorías Académicas de la Universidad Técnica de Ambato.

Inicio de Sesión



**TUTORÍAS
ACADÉMICAS**

Iniciar sesión

✉ Correo electrónico

🔒 Contraseña

Ingresar

El docente o estudiante debe digitar su correo institucional completo y la contraseña.

Funcionalidades para estudiantes

Menú principal



i

Cuando el estudiante inicie sesión, se redirigirá al menú como las funcionalidades que podrá ejecutar:

- Solicitar tutorías.
- Registrar tutorías (validarlas).
- Ver estados de las tutorías.

Solicitar Tutorías Académicas: Para poder solicitar una tutoría, el estudiante debe considerar las siguientes opciones.

The screenshot shows a mobile application interface for requesting academic tutoring. At the top, there is a red header with a back arrow and the text 'Solicitar tutoría'. Below the header are three white input fields: 'Tema' with a red 'Tr' icon, 'Materia' with a dropdown arrow and the text 'Seleccione un...', and 'Horario' with a dropdown arrow and the text 'Seleccione un...'. At the bottom of the form is a red button labeled 'Solicitar'. Three blue arrows point from the 'Tema', 'Materia', and 'Horario' fields to corresponding text boxes on the right.

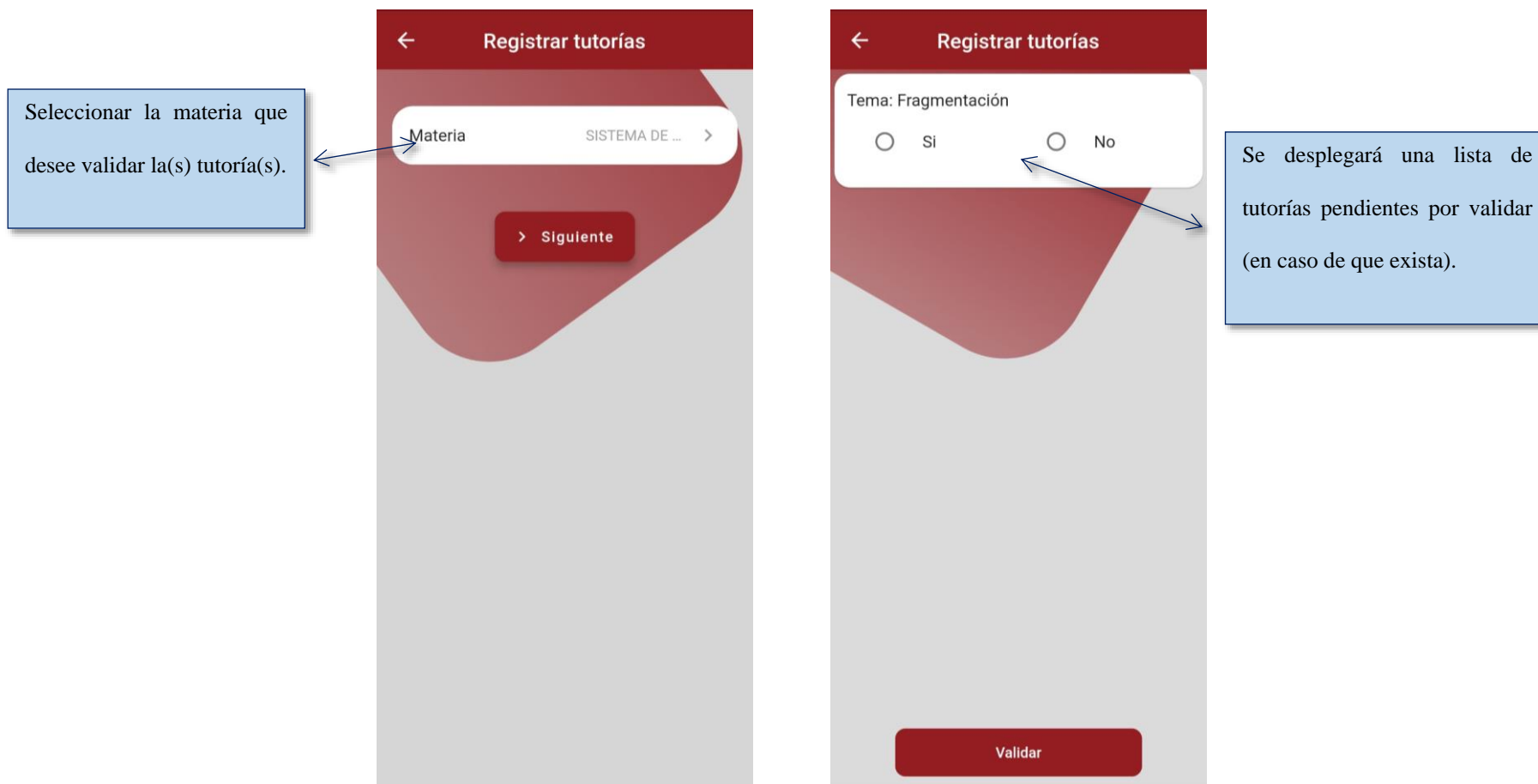
Tema a tratar.

Seleccionar la materia que desee recibir la tutoría.

El estudiante debe seleccionar el horario para la tutoría.

- i**
- Solo aparecerán las materias donde el estudiante esté matriculado actualmente.
 - Los horarios se cargarán acorde a la materia seleccionada.
 - Luego de 1 hora, le llegará la notificación de que finalizó la tutoría para que pueda validarla.

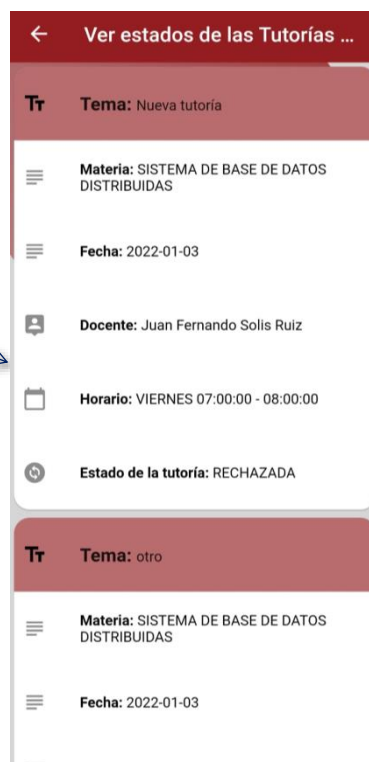
Registrar Tutorías Académicas: Para poder registrar una tutoría, el estudiante debe considerar las siguientes opciones.



Ver estados de las Tutorías Académicas: Para poder ver los estados de las tutorías, el estudiante debe considerar las siguientes opciones.

Los datos que el estudiante podrá visualizar son:

- Tema de la tutoría.
- Materia solicitada o registrada.
- Fecha de solicitud o de registro.
- Docente que imparte la materia.
- Horario de la tutoría.



El estudiante debe seleccionar la materia como en las anteriores funcionalidades.

Funcionalidades para docentes

Menú principal



i

Cuando el docente inicie sesión, se redirigirá al menú como las funcionalidades que podrá ejecutar:

- Registrar tutorías.
- Gestionar Tutorías Académicas.
- Ver estados de las Tutorías Académicas.

Registrar Tutorías Académicas: Para poder registrar una tutoría, el docente debe considerar las siguientes opciones.

Tema a tratar.

El docente debe seleccionar la materia donde tutoró a los alumnos.

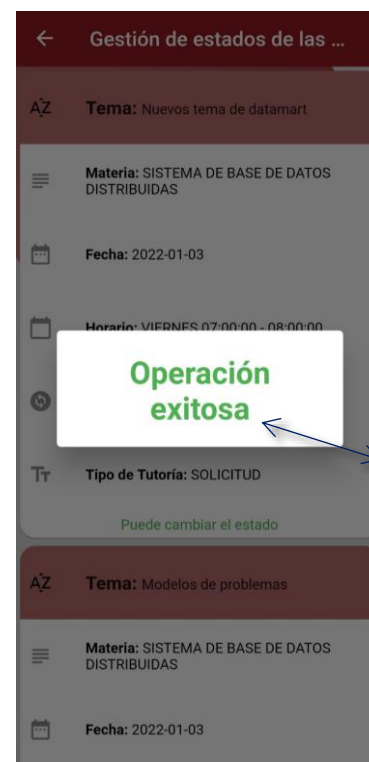
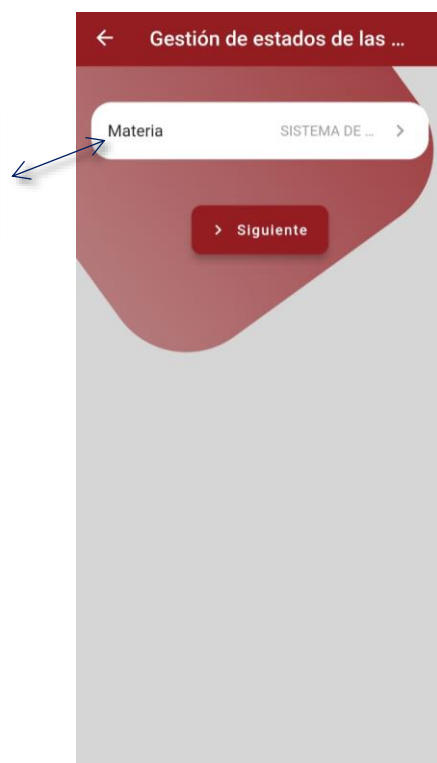
El docente debe seleccionar el horario en el que se dio la tutoría.

Cuando de click en siguiente, le aparecerá la siguiente ventana con el listado de los estudiantes matriculados en dicha materia.

Solo debe seleccionar a los estudiantes y dar click en registrar.

Gestionar de Tutorías Académicas: Para poder gestionar las tutorías, el docente debe considerar las siguientes opciones.

Seleccionar la materia que desee para poder gestionar las tutorías.



Al dar click en siguiente, al docente le aparecerá una lista con las tutorías y podrá modificar el estado de las mismas siempre y cuando cumpla con las condiciones:

- El horario en el que esté la solicitud sea en actividades académicas.
- Solo puede rechazar solicitudes.
- El estado de la tutoría se encuentra como PENDIENTE.

i El docente solo podrá rechazar una solicitud si este mensaje.

Ver estados de las Tutorías Académicas: Para poder ver los estados de las tutorías, el docente debe considerar las siguientes opciones:

Seleccionar la materia que desee para poder ver las tutorías.

- Las opciones son:
- Ver todos.
 - Ver las tutorías validadas.
 - Ver las tutorías no validadas.
 - Ver las tutorías pendientes



Al dar click en Siguiete, aparece una nueva ventana con la lista de las tutorías y el estado