

“AsesoraMe”, una aplicación móvil para la gestión de tutorías académicas, un enfoque desde la Ingeniería de Software

“AsesoraMe”, a Mobile Application for the Management of Academic Tutoring, an Approach from Software Engineering

Iker Antonio Pluma Amaro¹, Lizeth Hernández Rodríguez²
Jesús Rojas Hernández³, Liliana Palacios Huerta⁴

¹Instituto Politécnico Nacional, Ciudad de México, MÉXICO
ORCID: 0009-0000-9815-1373 | iplumaa2100@alumno.ipn.mx

²Instituto Politécnico Nacional, Ciudad de México, MÉXICO
ORCID: 0009-0006-7691-2892 | lhernandezr2103@alumno.ipn.mx

³Instituto Politécnico Nacional, Ciudad de México, MÉXICO
ORCID: 0009-0002-0120-3347 | jrojashe@ipn.mx

⁴Instituto Politécnico Nacional, Ciudad de México, MÉXICO
ORCID: 0000-0002-6200-300X | lpalacios@ipn.mx

Recibido 31-08-2023, aceptado 30-01-2024.

Resumen

En este artículo se reporta la creación de “AsesoraMe” una aplicación móvil diseñada para agilizar y potenciar la coordinación de tutoría entre los estudiantes y profesores de la Unidad Profesional Interdisciplinaria de Ingeniería Campus Tlaxcala (UPIIT), perteneciente al Instituto Politécnico Nacional. La asesoría entre pares es una figura tutorial en IPN que tiene como finalidad la regularización de estudiantes y prevención de la deserción estudiantil, suceso recurrente en el nivel superior. El cimiento AsesoraMe se erige en la capacidad brindada a los alumnos para registrarse, crear perfiles y especificar sus áreas de conocimiento y disponibilidad con el propósito de ofrecer tutoría o bien, decidir el rol a tomar (alumno, asesor o ambos). Además, la plataforma incorpora un sistema de programación de citas y recordatorios asegurando la puntualidad y eficacia de las sesiones, así como explorar las opciones de asesoría disponibles y aplicar filtros acordes a sus necesidades específicas.

La aplicación móvil fue planeada, diseñada e implementada a través de la ingeniería de software, esta disciplina se interesa por todos los aspectos de la producción de software, tiene la ventaja que se ocupa desde las primeras etapas de especificación del sistema hasta el mantenimiento del sistema después de que se pone en operación. Las tecnologías empleadas para el desarrollo de la aplicación fueron Dart, JavaScript, MySQL.

Palabras clave: tutorías académicas, aplicación móvil, ingeniería de software.

Abstract

This article reports the creation of “AsesoraMe” a mobile application designed to expedite and enhance the coordination of tutoring between students and teachers of the Unidad Profesional de Ingeniería Campus Tlaxcala (UPIIT), part of the Instituto Politécnico Nacional. Peer counseling is a tutorial figure in IPN that aims to regularize students and prevent student desertion, a recurring event at the higher level. The foundation of AsesoraMe is the ability provided to students to register, create profiles and specify their areas of knowledge and availability in order to offer tutoring or deciding the role to take (student, advisor or both). In addition, the platform incorporates an appointment and reminder scheduling system ensuring the timeliness and effectiveness of the sessions as well as exploring the available advisory options and applying filters according to their specific needs.

The mobile application was planned, designed and implemented through software engineering; this discipline is interested in all aspects of software production. It has the advantage that it deals from the early stages of system specification to the maintenance of the system after it is put into operation. The technologies used for the development of the application were Dart, JavaScript, MySQL.

Index terms: academic tutoring, mobile application, software engineering.

I. INTRODUCCIÓN

En el contexto educativo actual, la coordinación efectiva de asesorías académicas se ha convertido en un aspecto clave para mejorar el rendimiento estudiantil y fomentar un entorno de aprendizaje colaborativo. La necesidad de brindar apoyo personalizado a los estudiantes en el proceso de formación ha llevado a la creación de soluciones innovadoras, como aplicaciones y páginas web diseñadas para optimizar la comunicación entre aquellos que buscan orientación y quienes están dispuestos a proporcionarla. Como por ejemplo Linc, una plataforma web y aplicación móvil cuyo propósito es automatizar el proceso de asesorías académicas, esta plataforma tiene varios planes que incluyen un costo por servicio. “Asesorías ESFM - Tutoría entre pares” es una página de facebook en cuyo perfil menciona que agenda asesorías, información y videos para resolver dudas de física y matemáticas, esta página pertenece a la Escuela Superior de Física y Matemáticas (ESFM).

En IPN, la tutoría entre pares y la innovación educativa son áreas prioritarias. Cada año la institución capacita a los profesores y profesoras en el ámbito de tutorías, donde se destaca la tutoría grupal y la tutoría entre pares como una herramienta de apoyo en la regularización de los estudiantes y prevención de la deserción estudiantil. Además, IPN cuenta con un Programa Institucional de Innovación Educativa (PIInE), el cual constituye un marco de referencia para el fomento e impulso de las Prácticas Educativas Innovadoras y con ello mejorar los procesos de enseñanza, aprendizaje y gestión educativa [1].

A pesar de los avances tecnológicos y las iniciativas implementadas, persisten desafíos recurrentes en la coordinación de asesorías en el ámbito universitario del IPN. Estos desafíos pueden tener un impacto negativo en la calidad de la educación y en la retención estudiantil. La falta de una plataforma efectiva que facilite la interacción entre estudiantes y asesores ha creado una brecha en la disponibilidad de recursos de apoyo [2].

En este escenario, este trabajo se centra en la planificación, diseño y desarrollo de la aplicación móvil AsesoraMe, que busca abordar estos desafíos y mejorar la coordinación de asesorías en la Unidad Profesional Interdisciplinaria de Ingeniería Campus Tlaxcala (UPIIT), perteneciente al Instituto Politécnico Nacional. A través de la implementación de esta aplicación, se pretende proporcionar a los estudiantes una aplicación móvil accesible y eficiente para acceder a asesorías personalizadas, mejorando la comunicación y colaboración entre estudiantes y tutores en el ámbito de la educación.

AsesoraMe es una aplicación móvil que busca establecer una base tecnológica sólida utilizando principios de ingeniería de software, debido a que esta es una disciplina que se interesa por todos los aspectos involucrados en la producción de software desde las primeras etapas de especificación del sistema hasta el mantenimiento del sistema posterior a su proceso operativo, dentro de sus ventajas se destaca que a menudo resulta más económico a largo plazo usar métodos y técnicas de ingeniería de software, que solo diseñar los programas como si fuera un proyecto de programación personal.

Para agilizar y optimizar el desarrollo de la propuesta se utilizó como base el ciclo de vida del desarrollo de software, que básicamente es un enfoque sistemático que guía la creación y evolución del software desde su concepción hasta su entrega y mantenimiento. Este ciclo consta de diversas fases interconectadas, la planificación, diseño, implementación y mantenimiento efectivo del software [3].

La Fig. 1 muestra las fases del desarrollo de software definidas por Pressman [3].



Fig. 1. Organizador gráfico que aborda las fases del ciclo de vida del software.

A continuación, se muestran algunos requerimientos funcionales que el sistema debe satisfacer:

- R1: El sistema permitirá a los usuarios crear cuentas de acceso a la aplicación.
- R2: Un usuario debe acceder al sistema a través de su cuenta.
- R3: Una cuenta debe tener al menos un rol, el cual puede ser alumno, asesor o ambos.
- R4: Un usuario con cuenta puede consultar asesorías programadas.
- R5: Un usuario con cuenta puede consultar actividades extracurriculares.
- R6: Un usuario con cuenta puede consultar asesorías personales pertenecientes a su cuenta.
- R7: Un usuario con rol alumno puede solicitar asesorías personales.
- R8: Un usuario con rol de asesor o ambos puede consultar asesorías solicitadas por otros usuarios.
- R9: Un usuario con rol de asesor o ambos puede ofertar asesorías personales.
- R10: Un usuario con rol de asesor puede agregar asesorías programadas.
- R11: Un usuario con rol de asesor puede crear actividades extracurriculares.
- R12: Un usuario con rol alumno puede aceptar propuestas de asesorías personales.
- R13: Un usuario con rol alumno debe poder calificar a otros usuarios.
- R14: Un usuario con cualquier rol puede ser calificado.
- R15: Un usuario puede modificar los datos de su cuenta.

Partiendo de los requerimientos funcionales se inició con la fase de diseño, en esta fase se tomaron en cuenta los detalles para generar estrategias de diseño que permitan sentar bases sólidas en las cuales el sistema se cimentará, la primera inquietud fue la de trabajar con objetos del mundo real -asesorías, alumnos, asesores-, como respuesta se decidió utilizar el paradigma de programación orientada a objetos, este paradigma de programación permite

crear programas con base en unidades llamadas objetos, estos objetos son entidades virtuales con datos y funciones que modelan a un objeto del mundo real [4], sin embargo, a medida que el sistema se volvía más complejo, fue esencial utilizar estrategias que ayudaran a manejar y resolver desafíos propios del crecimiento del sistema, por eso mismo, se utilizaron patrones de diseño para facilitar la resolución de estos desafíos; un patrón de diseño describe un problema que se repite una y otra vez en un contexto específico, y también describe el núcleo de la solución a ese problema, de una manera en la que se pueda utilizar un millón de veces sin hacerlo de la misma forma dos veces [5].

4

El primer patrón utilizado entró en juego para facilitar la creación de los objetos previamente mencionados, el patrón de diseño de composición, este patrón permite definir herencias de objetos primitivos que pueden ser "compuestos" en objetos mucho más complejos [6], este patrón se empleó principalmente para modelar los diferentes tipos de asesorías, los demás patrones utilizados serán mencionados después.

Para la estructura del sistema se implementó el patrón MVC (Modelo Vista Controlador), este patrón consiste en 3 capas que separan las responsabilidades y permiten que trabajen de manera independiente unas de otras para así aumentar la modularidad del sistema, la primera capa es la capa de modelo esta capa, contiene la representación interna de los datos y define cómo se accede y se manipula esa información; la segunda capa es la capa de vista, esta capa es responsable de la presentación de los datos al usuario, la última capa es la capa de controlador: recibe las interacciones del usuario desde la vista y las traduce en acciones que afectan al modelo [7], las capas fueron implementadas en MySQL, que es un sistema de gestión de bases de datos relacional [8]; Dart, que es un lenguaje de programación del lado de cliente para desplegar aplicaciones en cualquier plataforma [9] y JavaScript ejecutado en NodeJS respectivamente, JavaScript siendo un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas [10] y NodeJS un entorno de ejecución de JavaScript del lado del servidor [11].

En el siguiente apartado, se describirán los métodos y enfoques utilizados en el diseño e implementación de la aplicación móvil, seguido de la presentación de los resultados obtenidos y su análisis. A través de este trabajo, se pretende aportar una solución práctica y relevante, sin tareas administrativas de por medio a un problema crucial en el ámbito educativo, resaltando la importancia de la coordinación de asesorías para el desarrollo académico de los estudiantes que son la primera prioridad del IPN.

II. METODOLOGÍA

Las fases de análisis, desarrollo y pruebas son componentes críticos e inseparables en el ciclo de vida del software. A medida que una idea abstracta evoluciona hacia un producto funcional, estas fases desempeñan papeles esenciales en la transformación de conceptos a aplicaciones tangibles y robustas. El contenido siguiente se encargará de explicar cómo el sistema fue evolucionando hasta llegar a cumplir con los requerimientos funcionales expresados con anterioridad, finalizando con las pruebas que avalen la satisfacción de las necesidades.

A. Casos de uso

Para modelar los comportamientos principales de la aplicación se realizó un diagrama de casos de uso, que permite modelar la funcionalidad de un sistema tal y como lo perciben los agentes externos, llamados autores, que interactúan con el sistema en un comportamiento específico [12].

La Fig. 2 modela un caso de uso específico referente a la interacción entre el usuario y las asesorías dentro del sistema, puede observarse la diferencia de permisos entre los diferentes tipos de usuario y las capacidades que tiene cada uno.

5



Fig. 2. Caso de uso sobre los requerimientos R4, R5, R6, R7, R8, R9, R10, R11 y R12.

La Fig. 3 muestra el caso de uso de manejo de cuentas de usuario, se puede observar cómo ambos tipos de usuarios aparece en la mayoría de los casos de uso, pero diferenciando a los usuarios de rol “Asesor” que es el único cuyo desempeño puede ser calificado.

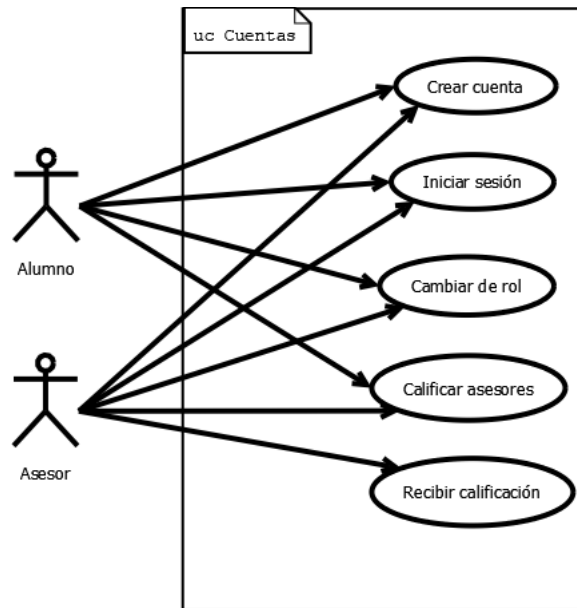


Fig. 3. Caso de uso sobre los requerimientos R1, R2, R3, R13, R14 y R15.

B. Estructura del sistema

Un diagrama de paquetes se utiliza principalmente para describir la estructura de un sistema completo mediante organización jerárquica de alguna propiedad del sistema [13], en los diagramas se presenta la organización del diagrama en vistas, cada vista contendrá componentes, el componente es una pieza modular de un sistema cuyo comportamiento externo es descrito de manera más concisa en su implementación [14], esto permite reflejar la estructura de alto nivel del sistema.

6

El diagrama de paquetes mostrado en la Fig. 4 representa los paquetes principales del sistema, el paquete de presentación contiene todos los componentes relacionados con la interfaz de usuario de la aplicación, además, se hace hincapié en el uso del patrón Singleton, este es un patrón se encarga de mantener solo una única instancia de clase en algún programa [15], que fue utilizado para la tener una única salida en consola al depurar el programa. La capa del controlador está modelada por el paquete Business y el paquete Data, en el primer paquete se encuentran los componentes encargados de recuperar la información y obtener como salida objetos instanciados que fueron utilizadas para su renderización en la capa de Presentación -clases presentadas en la Fig. 5-. En este paquete también se remarcó el uso del patrón Singleton, ya que nos sirvió para tener una única conexión con la API -REST- correspondiente, API REST es un estilo de arquitectura de software que permite la comunicación y la transferencia de datos entre sistemas de manera simple y eficiente para realizar operaciones en recursos identificados por los URL [16]. El paquete Data contiene las API utilizadas por el sistema, cada una de estas API tiene un propósito definido que ofrece únicamente datos de su tabla especificada. Finalmente, la capa de Modelo se basó en el paquete Database, donde se encuentra nuestro único componente que es la conexión y obtención de datos de la base de datos utilizada.

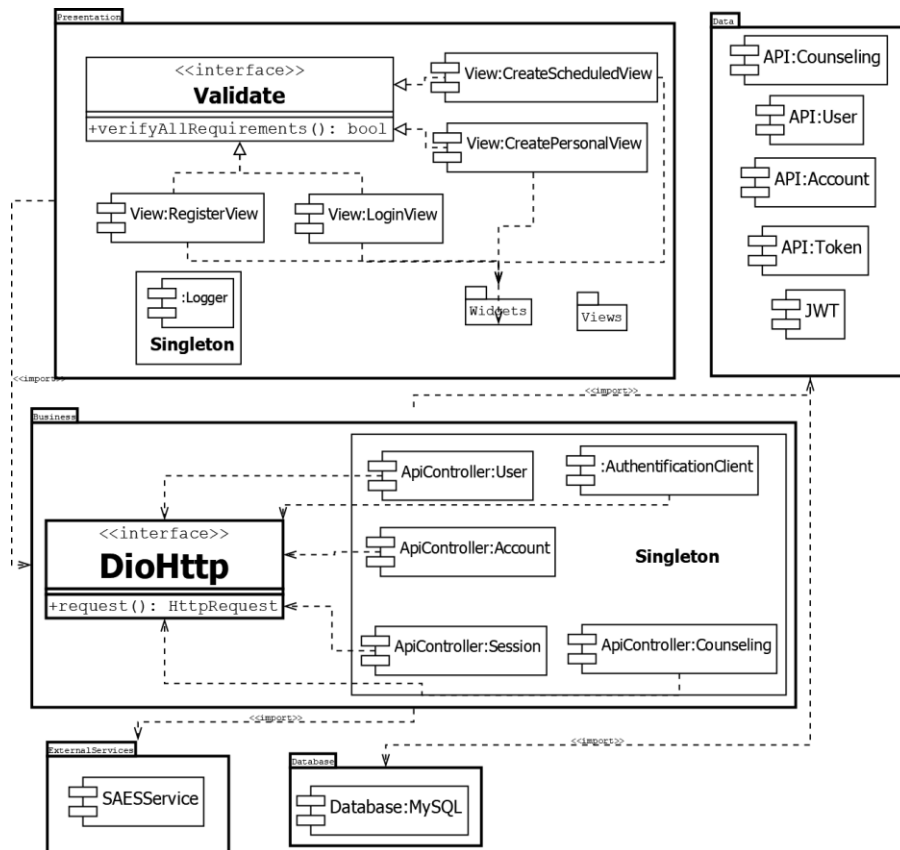


Fig. 4. Diagrama de paquetes que modela el funcionamiento de la aplicación.

La Fig. 5 muestra la estructura propuesta que modela las clases utilizadas en la aplicación del lado del Front-end o la vista, como se puede observar, debido al patrón utilizado, el funcionamiento de esta capa de abstracción sirvió principalmente para el muestreo de información.

Para la creación de las clases utilizamos el patrón de composición, esto con el propósito de crear clase con la composición de otras clases, aunque el mayor grado de herencia en esta propuesta es de solo 1 generación, este patrón nos permitió la separación de clases de manera semántica.

7

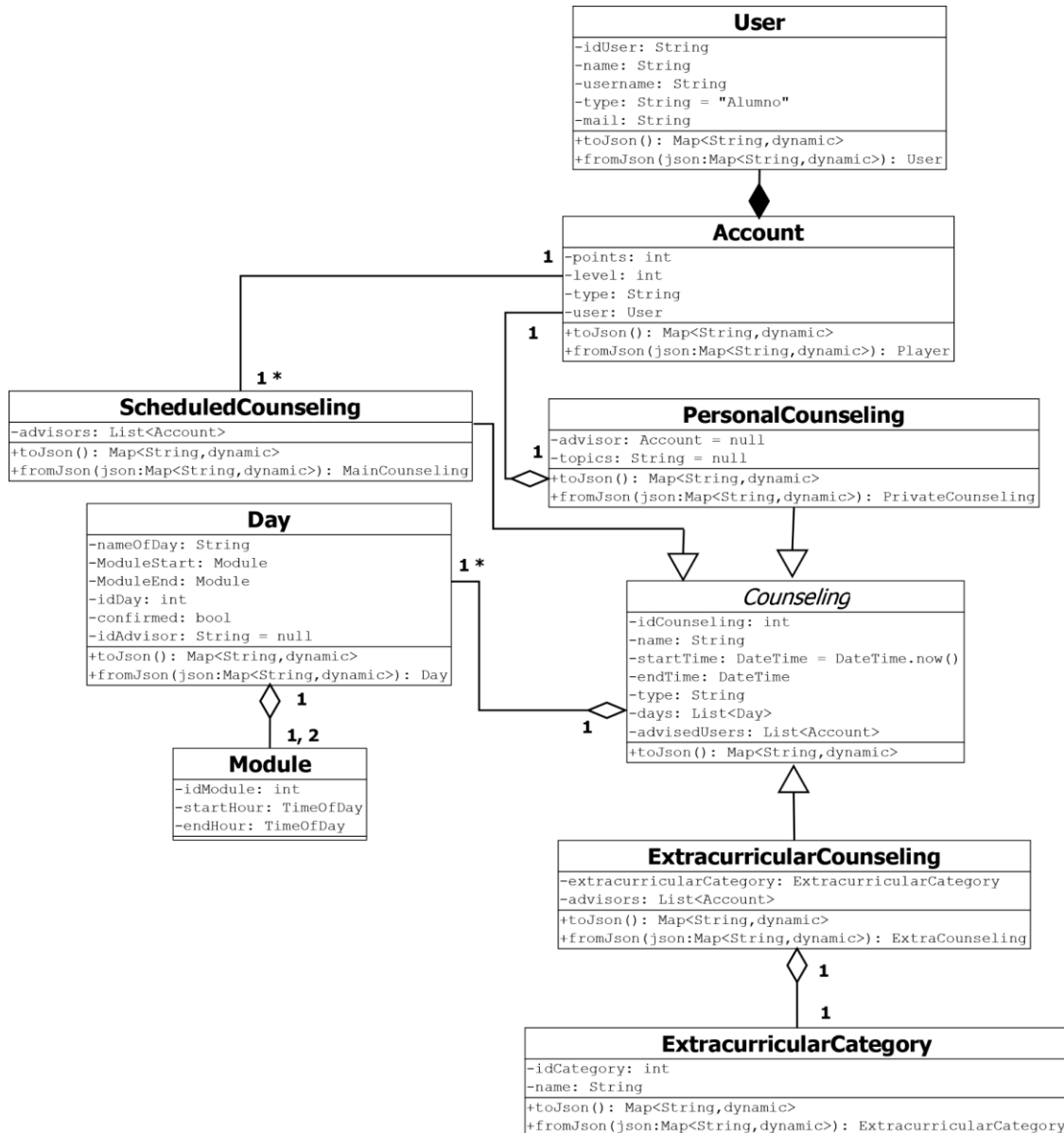


Fig. 5. Diagrama de clases que define la estructura del Front-end.

El diagrama de secuencias nos permite representar el tiempo en el que las clases de nuestro sistema interactúan y la cronología con lo que lo hacen [17], por esto se utilizó para modelar la obtención de asesorías personales y como estas son renderizadas en el GUI de la aplicación. La Fig. 6 muestra el diagrama de secuencias para este caso particular, como primer lapso de tiempo, el usuario ingresa en la página de asesorías personales, una vez dentro, el sistema trabaja internamente mediante una petición GET al URL específico de la API, la API hace una petición a la base de datos, la primera petición recibe una lista de enteros que serán los ID de las asesorías personales (cabe recalcar que las asesorías son las que pertenecen al usuario que hizo la petición), después, por cada uno de los ID recuperados, se hacen peticiones correspondientes para obtener toda la información de las asesorías, finalmente, se renderizan las asesorías en sus respectivos widgets, que además de mostrar la información de manera ordenada le dará funcionalidad extra.

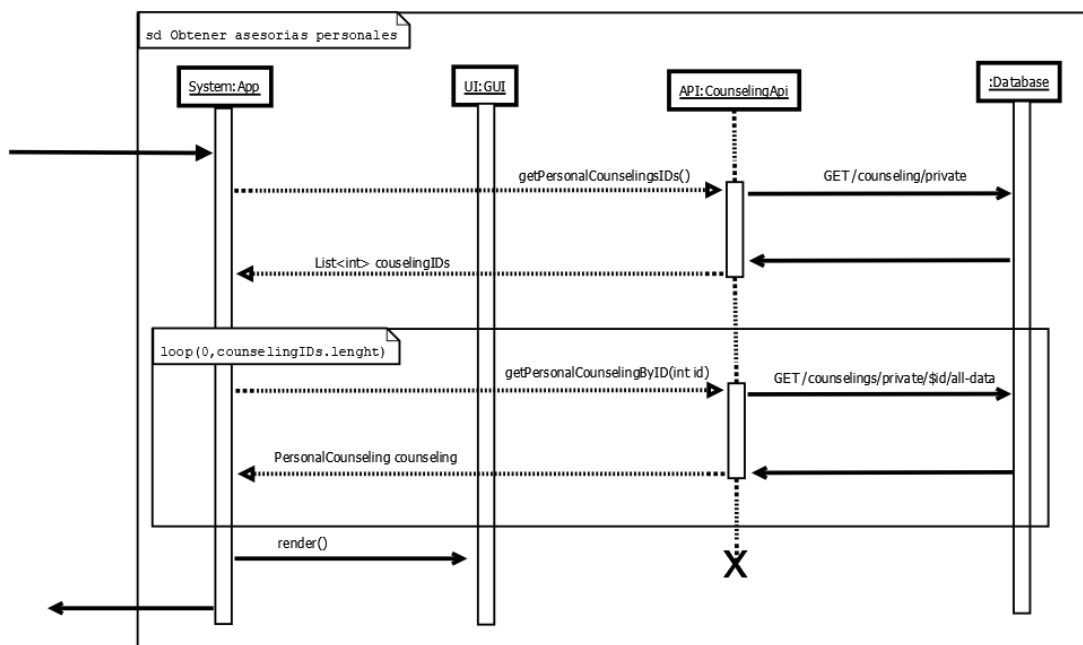


Fig. 6. Diagrama de secuencias para la obtención de asesorías personales y su renderización en el GUI.

A. Demostración de funcionalidad

1) API

Para la creación de la documentación de la API [18] se utilizó el programa POSTMan, que es un programa para la creación y testeo de las API [19]. Por motivos de la seguridad del usuario se implementó el almacenamiento de contraseñas encriptadas utilizando la librería "bcryptjs" de JavaScript, que es una librería que permite encriptar cadenas de texto [20].

Para la obtención de información sobre las asesorías se implementó una ruta específica para cada tipo de asesoría, en la Fig. 7 se muestra el endpoint para obtener información sobre una actividad extracurricular, los tres endpoints de asesorías siguen la misma estructura /counselings/tipo/id/all-data.

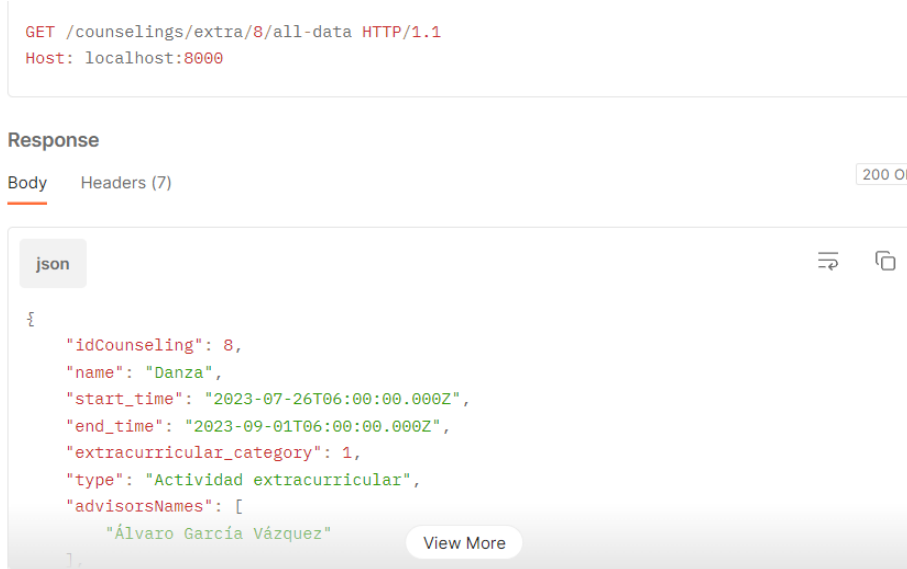


Fig. 7. Diagrama de Documentación de API: Obtención de información de actividad extracurricular específica.

2) Registro de un usuario con rol de "alumno" e inicio de sesión del mismo

La Fig. 8 muestra el flujo principal para la creación e inicio de sesión de una cuenta desde la aplicación, las vistas están conformadas de simples formularios para hacer el registro de los datos ingresados por el cliente, para registrar a un usuario es necesario llenar las entradas de texto y presionar el botón "Registrarse", los requerimientos para crear una cuenta son el tamaño de la contraseña o el nombre de usuario, entre otros, el botón de "Registrarse" se activará únicamente cuando todos los requerimientos sean satisfechos; el inicio de sesión funciona de manera similar.

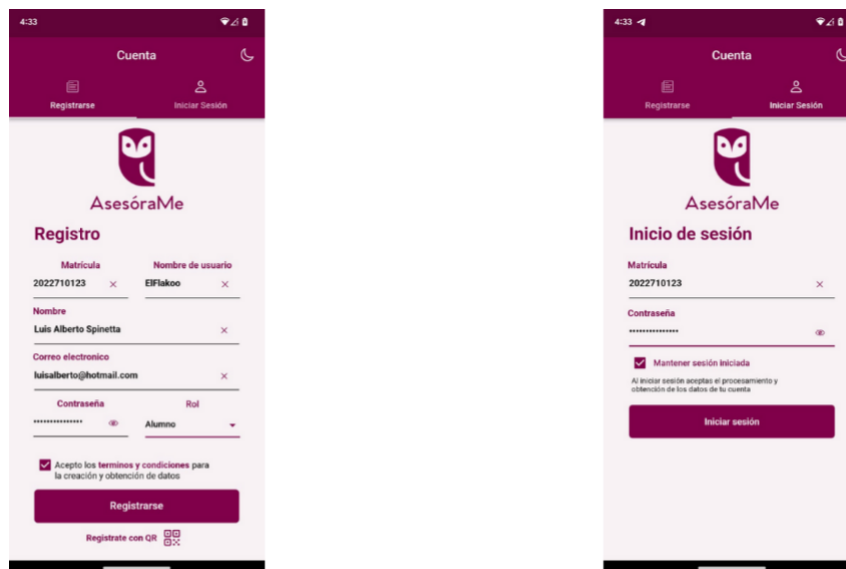


Fig. 8. Vistas para el flujo principal de registro e inicio de sesión dentro de la aplicación.

3) Registro de un usuario mediante QR

Como flujo alternativo para el registro de una cuenta se implementó la funcionalidad de recuperar los datos principales de un usuario con ayuda del código QR que se encuentra al reverso de las credenciales de los alumnos de la universidad, para seguir este flujo se selecciona el botón “Regístrate con QR” y se abrirá una ventana nueva, al colocar el código que se encuentra al reverso de las credenciales de los alumnos, se generarán los datos correspondientes a la matrícula, nombre y se generará un nombre de usuario aleatorio

4) Creación de asesoría pública

Para generar una asesoría programada (pensado para aquellas asesorías controladas por la unidad académica) el usuario deberá seleccionar la última pestaña de la página principal, en esta pestaña (similar a la mostrada en la primera vista de la Fig. 9). El usuario deberá presionar el botón “+” (solo disponible para usuarios con rol de tipo “Asesor” o “Ambos”), una vez presionado se abrirá una ventana similar a la mostrada en la primera vista de la Fig. 9. En esta ventana el usuario es libre de crear una asesoría pública o una actividad extracurricular, siguiendo los parámetros de la asesoría entre pares. Una asesoría puede ser impartida por más de un asesor, el usuario puede agregar asesores presionando el botón “+” o eliminando el último asesor agregado si presiona el botón “-”, el asesor seleccionará los módulos en los cuales la asesoría será impartida y el inicio y final de esta (debido a la formalidad de estas asesorías el registro de estas fechas es obligatorio), las fechas pueden ser ingresadas mediante la ayuda de la ventana modal que se abrirá al presionar el campo.

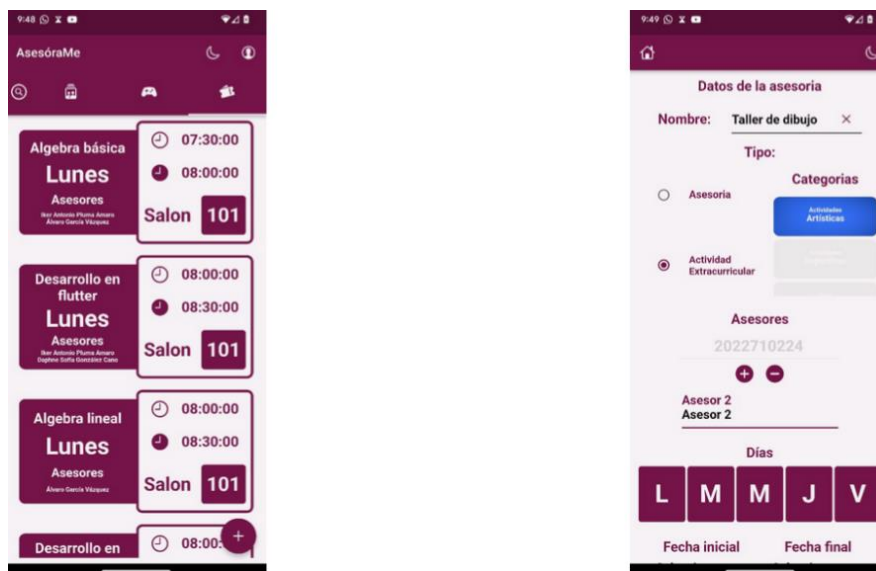


Fig. 9. Vistas del flujo principal para la creación de asesorías programadas.

5) Creación de una petición de asesoría personal

Para crear una petición de asesoría personal el usuario deberá seleccionar la primera (o segunda, en caso de que la cuenta pertenezca a un “Asesor” o “Ambos”) pestaña de la página principal, mostrada en la primera vista de la Fig. 10, en esta página deberá presionar el botón que se encuentra en la parte inferior derecha, se abrirá una vista parecida a la mostrada en la segunda vista de la Fig. 10, en donde el usuario se encargará de llenar los datos necesarios. Se añadió un campo opcional de temas para limitar los temas de la asesoría, al momento de

seleccionar los días con los que dispone el asesorado para tomar asesorías se abrirá una ventana modal donde podrá seleccionar el final e inicio de cada periodo, finalmente, el usuario presionará el botón “Crear” y la solicitud se publicará en el apartado de los asesores.

11

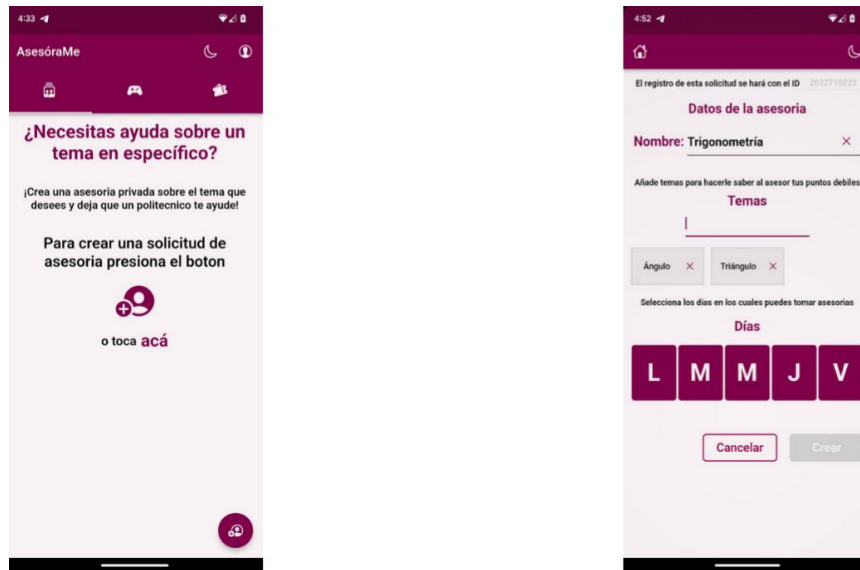


Fig. 10. Vistas del flujo principal para la creación de peticiones de asesorías personales.

6) *Generar propuesta de asesoría para una asesoría personal*

Para generar una propuesta para asesorar una asesoría personal el usuario deberá tener una cuenta con rol “Asesor” o “Ambos”. Si el usuario cuenta con alguno de estos roles tendrá una ventana más que se encuentra en la primera posición, donde se mostrarán asesorías personales que aún no tienen asesor asignado, el usuario deberá seleccionar (presionar) alguna de estas asesorías y proponer los días y horarios en los que puede dar clases al asesorado como paso final el usuario presionará el botón de “Crear” y la propuesta será creada

7) *Aceptar solicitud de una asesoría personal*

El usuario tiene la capacidad de seleccionar a su asesor en relación al tiempo que imparta o los días que seleccione, en la primera vista de la Fig. 11 se muestra una carta con propuesta de asesoramiento. El usuario seleccionará la asesoría con las peticiones y se abrirá la una nueva vista donde renderizan las propuestas acompañadas de una barra verde que será proporcional al tiempo de asesoramiento por día y de manera semanal, con la finalidad de aumentar la accesibilidad y concentrar información importante, en esta vista el usuario tiene la libertad de aceptar la propuesta que más le convenga, se genera una ventana modal para confirmar la selección y después se actualiza la ventana de asesorías personales del usuario.

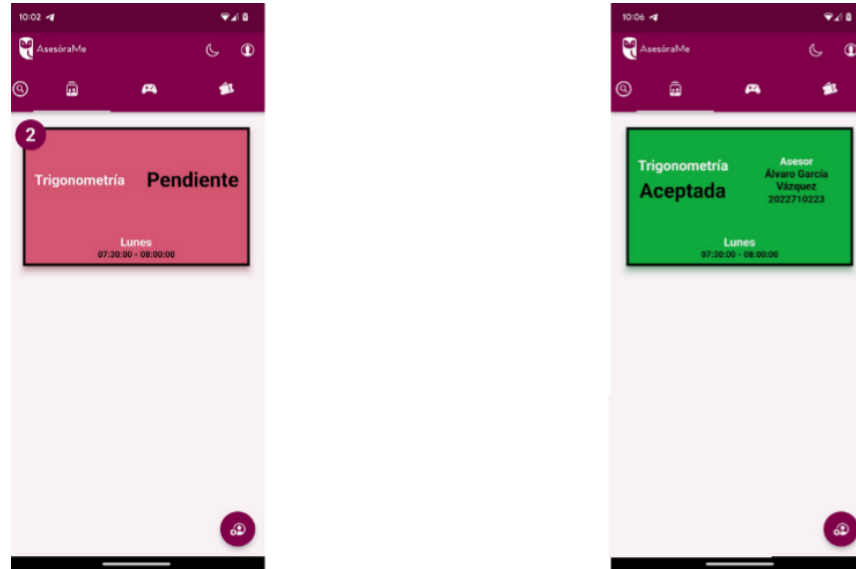


Fig. 11. Vistas del flujo principal para la aceptación de propuestas para una asesoría personal específica.

III. CONCLUSIONES

La aplicación de la ingeniería de software en el desarrollo de AsesoraMe fue parte fundamental para garantizar la creación de un producto eficiente, confiable y adecuado a las necesidades planteadas por el usuario. La ingeniería de software proporciona un enfoque estructurado y metodológico para el diseño, desarrollo, implementación y mantenimiento de la aplicación, lo que resulta en una serie de beneficios significativos. De esta manera aseguramos la calidad del producto final al enfocarse en la planificación, el diseño detallado y la codificación bien estructurada. Esto se traduce en una aplicación que es funcional, fácil de usar y libre de errores. Además, el enfoque en la ingeniería de software permite una mayor escalabilidad, lo que significa que la aplicación puede adaptarse y crecer según las necesidades cambiantes de la institución educativa y los usuarios.

Al seguir un enfoque estructurado, involucrar a expertos multidisciplinarios y priorizar la calidad del producto, se busca crear una aplicación que no solo satisfaga las necesidades educativas, sino que también contribuya positivamente a la experiencia de aprendizaje y enseñanza. La integración de tecnología en las asesorías académicas representa un paso crucial hacia la mejora de la calidad y la eficacia de la educación. Al aprovechar las herramientas y plataformas tecnológicas disponibles, se enriquece la experiencia de aprendizaje tanto para los estudiantes como para los asesores, brindando beneficios significativos.

La aplicación de patrones de diseño y la utilización de software libre en la generación de un proyecto representan un enfoque estratégico y efectivo para desarrollar soluciones tecnológicas de alta calidad, eficiencia y sostenibilidad, además de ofrecer un marco probado y consolidado para abordar desafíos comunes en el desarrollo de software. Estos patrones encapsulan las mejores prácticas y soluciones a problemas recurrentes, lo que resulta en una arquitectura más coherente, mantenible y escalable. Al utilizar patrones, se fomenta la reutilización de soluciones exitosas y se evitan errores comunes, lo que acelera el proceso de desarrollo y mejora la calidad del producto final.

El desarrollo de software alineado al Programa Institucional de Innovación y Desarrollo (PIIne) del Instituto Politécnico Nacional (IPN) representa un paso significativo hacia la materialización de la visión y los objetivos institucionales de promover la investigación, la innovación y el desarrollo tecnológico. AsesoraMe es un proyecto de software que se integra de manera armónica con las áreas prioritarias del PIIne, logrando una simbiosis que refuerza la capacidad del IPN para generar soluciones impactantes y vanguardistas.

Este enfoque asegura que el desarrollo de software no sea simplemente una respuesta técnica a una necesidad puntual, sino una iniciativa estratégica que nutre el entorno académico, científico y social del IPN. Mediante la colaboración multidisciplinaria y la aplicación de ingeniería de software.

La integración de un enfoque innovador, la atención a la calidad, la generación de conocimiento y la transferencia de tecnología son elementos esenciales para el éxito de un proyecto de software. AsesoraMe dirige sus esfuerzos hacia desafíos significativos, transformando problemas en oportunidades y resultados en soluciones tangibles. La innovación y la investigación, respaldadas por un enfoque de desarrollo de software bien estructurado, permiten a la institución avanzar hacia la vanguardia del conocimiento y contribuir de manera significativa al desarrollo sostenible de la sociedad en su conjunto.

También es preciso remarcar, que AsesoraMe usa tecnologías libres y se oferta también como software libre, acercando esta tecnología a instituciones y usuarios que no cuentan con los recursos suficientes para pagar por un software de este tipo. La adopción de software libre aporta múltiples ventajas, incluyendo la reducción de costos, la flexibilidad y la colaboración comunitaria. Al aprovechar herramientas y bibliotecas de código abierto, se evita la necesidad de desarrollar componentes desde cero y se acelera el desarrollo. Además, la comunidad de software libre brinda soporte continuo, actualizaciones y mejoras, lo que contribuye a la longevidad y la calidad del proyecto.

REFERENCIAS

- [1] Instituto Politécnico Nacional. “Portal del Instituto Politécnico Nacional”. IPN.mx. Accessed Aug. 30, 2023. Available: <https://www.ipn.mx/dfie/innovaci%C3%B3n/practic-as-educativas.html>
- [2] V. Tinto, “Dropout from higher education: A theoretical synthesis of recent research”, *Rev. Educ. Res.*, vol. 45, n.o 1, pp. 89-125, 1975, doi: <https://doi.org/10.3102/0034654304500108>
- [3] R. S. Pressman, *Ingeniería del software: un enfoque práctico*, 5ª ed., México: McGraw-Hill, 2014.
- [4] J. Cervantes, et al., *Introducción a la programación orientada a objetos*, 1ª ed., México: Universidad Autónoma Metropolitana, 2016.
- [5] E. Gamma, et al., “What is a design pattern?” in *Design patterns: Elements of reusable object-oriented software*, 1ª ed., USA: Addison-Wesley Professional, 2008, pp. 12-13.
- [6] E. Gamma, et al., “Structural patterns: Compose” in *Design patterns: Elements of reusable object-oriented software*, 1ª ed., USA: Addison-Wesley Professional, 2008, pp. 183-195.
- [7] R. D. Hernández. “The model view controller pattern – MVC architecture and Frameworks explained”. freeCodeCamp.org. Accessed Aug. 24, 2023. Available: <https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/>
- [8] IONOS. “¿Qué es mysql?, IONOS Digital Guide”. Ionos.mx. Accessed Aug. 24, 2023. Available: <https://www.ionos.mx/digitalguide/servidores/know-how/que-es-mysql/>
- [9] Dart. “An approachable, portable, and productive language for high-quality apps on any platform”. Dart.dev. Accessed Aug. 27, 2023. Available: <https://dart.dev/>
- [10] Amazon. “¿Qué es JavaScript?”. Amazon.com. Accessed Aug. 27, 2023. Available: <https://aws.amazon.com/es/what-is/javascript/>
- [11] Node.js. “Node.js”. Nodejs.org. Accessed Aug. 27, 2023. Available: <https://nodejs.org/es>
- [12] J. Rumbaugh, I. Jacobson, G. Booch, “Vista de casos de uso” in *El Lenguaje Unificado de modelado: Manual de Referencia*, 2ª ed., México: Pearson Addison Wesley, 2007, pp. 31-33.
- [13] R. Alarcón, “Paquetes” in *Diseño orientado a objetos con UML*, Madrid, España: Grupo EIDOS, 2000, pp. 37-38.
- [14] J. Rumbaugh, I. Jacobson, G. Booch, “Componente” in *El Lenguaje Unificado de modelado: Manual de Referencia*, 2ª ed., México: Pearson Addison Wesley, 2007, pp. 67-68.
- [15] D. Nesteruk, “Singleton” in *Design patterns in modern C++20: Reusable approaches for object-oriented software design*, 2ª ed., New York, USA: Apress, 2022, pp. 101-103.
- [16] F. Doglio, “REST 101” in *Rest API development with node.js: Manage and understand the full capabilities of successful rest development*, New York, USA: Apress, 2018, p. 8.
- [17] J. Schuller, et al., “Diagrama de Secuencias” in *Aprendiendo UML EN 24 Horas*. México: Prentice Hall, 2000, pp. 103-107.
- [18] I. Pluma, *Documentación de la API de AsesoraMe*, México: Instituto Politécnico Nacional, 2023.
- [19] Postman. “What is postman? postman api platform, Postman API Platform”. Postman.com. Accessed Aug. 27, 2023. Available: <https://www.postman.com/product/what-is-postman/>
- [20] H. Akhtar. “Bcryptjs, npm”. Npmjs.com. Accessed Aug. 27, 2023. Available: <https://www.npmjs.com/package/bcryptjs>