Ghilino, Ramiro

Villareal, Juan Manuel

Sistema informático para la gestión del programa de Tutorías de Acompañamiento Integral a Estudiantes (TAIE)

Tesis para la obtención del título de grado de Ingeniero de Sistemas

Directores:

Carreño, Ignacio Luciano

Porrini, Federico

Facultad de Ingeniería

Carrera de Ingeniería de Sistemas

Documento de Trabajo Final

Turnero TAIE

Sistema informático para la gestión del programa de Tutorías de Acompañamiento Integral a Estudiantes (TAIE)

Autores

Ramiro Ghilino

Juan Manuel Villarreal

Tutores

Ing. Ignacio Luciano Carreño

Ing. Federico Porrini

2024

Índice

Resumen

Las universidades son esenciales para la formación académica, pero persisten en ellas modelos de enseñanza tradicionales que pueden no ser efectivos para abordar las dificultades de los estudiantes, especialmente en los primeros años. Una solución a estas dificultades es la técnica de "knowledge sharing" o compartir conocimiento entre estudiantes, a través de tutorías pares, donde alumnos avanzados que ya han transitado y resuelto estas dificultades ayudan a los ingresantes.

El Programa de Tutorías de Pares (PTP) se implementó en la Universidad Católica de Córdoba (UCC) para mejorar la inserción y permanencia de los estudiantes. Posteriormente, surgió el Programa de Tutorías de Acompañamiento Integral a Estudiantes (TAIE), con el objetivo de unificar y mejorar estos esfuerzos.

El TAIE establece requisitos claros para los tutores, define áreas de intervención y propone un proceso de selección y seguimiento. Sin embargo, surgen desafíos en la implementación, como la diversidad de procesos de inscripción, limitaciones en el sistema de feedback y la falta de una plataforma centralizada para coordinar las tutorías.

Para abordar estas deficiencias, se propone desarrollar un sistema para mejorar la eficiencia y el funcionamiento del programa TAIE. A partir de la implementación del mismo, es posible una gestión más efectiva de las tutorías, una mejor comunicación entre tutores y alumnos, y una recopilación más completa de datos para evaluar el programa.

Este documento proporciona una descripción detallada del sistema propuesto "Turnero TAIE", que incluye sus características, implementación y componentes funcionales.

Abstract

Universities are essential for academic formation, yet they still harbor traditional teaching models that may not effectively address students' difficulties, particularly in their early years. One solution to these challenges is the technique of "knowledge sharing" or peer tutoring, where advanced students who have already navigated and resolved such difficulties assist incoming students.

The Peer Tutoring Program (PTP) was implemented at the Universidad Católica de Córdoba (UCC) to enhance students' integration and retention. Subsequently, the Comprehensive Student Support Tutoring Program (TAIE) emerged with the aim of consolidating and improving these efforts.

TAIE establishes clear requirements for tutors, defines intervention areas, and proposes a selection and monitoring process. However, challenges arise in implementation, such as diverse enrollment processes, feedback system limitations, and the lack of a centralized platform to coordinate tutoring sessions.

To address these shortcomings, the proposal suggests developing a mobile application, a web dashboard, and a database to enhance the efficiency and operation of the TAIE program. These modules would enable more effective tutoring management, better communication between tutors and students, and comprehensive data collection for program evaluation.

This document provides a detailed description of the proposed "TAIE Scheduler" system, encompassing its features, implementation, and functional components.

Glosario

Área de conocimiento: Un campo específico de estudio dentro de una o más asignaturas.

Android: Un sistema operativo móvil basado en Linux diseñado principalmente para dispositivos inteligentes, como teléfonos inteligentes y tablets.

API: Application Programming Interface, es un conjunto de reglas y especificaciones que permite que diferentes aplicaciones de software interactúen entre sí.

Backend: La parte de una aplicación o sitio web que maneja la lógica de negocio, bases de datos, modelos y comunicación con servidores.

Ciclo básico: Refiere a los primeros años de cursado en una carrera de grado.

Cliente: Se refiere a un dispositivo o sistema que solicita, consume y utiliza los recursos y servicios ofrecidos por un servidor.

Clave UCC: Identificador académico único para cada estudiante de la Universidad Católica de Córdoba.

CRM: Customer Relationship Management. Un sistema que gestiona las relaciones con los clientes de una empresa.

Código abierto: Software cuyo código fuente está disponible públicamente, lo que permite a cualquier persona utilizarlo, modificarlo y distribuirlo.

DOM: Document Object Model, es una representación estandarizada de un documento HTML o XML, permitiendo que a través de la programación se manipule el contenido, la estructura y el estilo del documento.

Experiencia de usuario: La calidad de la interacción que un usuario tiene con un producto o servicio, incluyendo la facilidad de uso y satisfacción general.

Feedback: Retroalimentación. Crítica sobre lo ocurrido. Utilizado para referir a la experiencia del tutor y el alumno luego de una instancia.

Framework: Un conjunto estructurado de componentes de software que facilitan el desarrollo de aplicaciones, proporcionando una base sobre la cual los desarrolladores pueden construir.

Frontend: La parte de una aplicación o sitio web que interactúa directamente con los usuarios, usualmente se refiere a la interfaz gráfica y experiencia del usuario.

HTTP: Hypertext Transfer Protocol, es el protocolo de transferencia de hipertexto utilizado para enviar y recibir datos en la web.

Interfaz de usuario: El punto donde los usuarios interactúan con una aplicación o sistema, generalmente a través de pantallas gráficas.

iOS: Un sistema operativo desarrollado por Apple para sus dispositivos móviles, como iPhones y iPads.

JSON: JavaScript Object Notation, es un formato ligero de intercambio de datos que es fácil de leer y escribir para humanos y fácil de parsear y generar para máquinas.

ORM: Object-Relational Mapping, es una técnica de programación que convierte los datos entre sistemas de tipos incompatibles en lenguajes de programación orientados a objetos.

PTP: Refiere a “Programa de Tutores Pares”

Repositorio: Refiere a un servidor donde se almacena y se administra el código fuente de un proyecto de desarrollo de software, o distintas bibliotecas y librerías de componentes.

RSU: Responsabilidad Social Universitaria. La RSU es aquella habilidad y efectividad de la Universidad para responder a las necesidades de transformación de la sociedad donde está inmersa, mediante el ejercicio de sus funciones sustantivas: docencia, investigación, extensión y gestión interna.

SDK: Software Development Kit, es un conjunto de herramientas de desarrollo utilizado para crear aplicaciones para un determinado entorno de ejecución o plataforma.

Servidor: Un sistema que proporciona recursos y servicios a otros computadores o sistemas, generalmente de manera remota.

SQL: Structured Query Language, es un lenguaje de programación utilizado para comunicarse con las bases de datos y manipular sus modelos e información.

TAIE: Refiere a “Tutorías de Acompañamiento Integral a Estudiantes”

Tutor: Estudiante encargado de brindar ayuda personalmente a un estudiante tutorando, y lo orienta en la planificación y ejecución de sus tareas académicas.

Tutorando: Es el estudiante que es guiado por un tutor en un proceso de aprendizaje.

Unidad académica: Refieren a la estructura académica de la Universidad, que se encuentra estructurada académicamente en diez Facultades, un Instituto de Ciencias de la Administración y un Departamento de Formación.

Widget: Un elemento de la interfaz de usuario que proporciona contenido o funcionalidad específica a una aplicación.

Presentación del tema

Este trabajo aborda la propuesta de mejora del programa universitario de Tutorías de Acompañamiento Integral a Estudiantes (TAIE) a través de la implementación de un sistema informático.

El programa se centra principalmente en estudiantes del ciclo básico. Esto se debe a los desafíos de inserción y adaptabilidad que pueden enfrentar estudiantes ingresantes al momento de unirse a la comunidad universitaria.

El programa TAIE fue implementado en todas las unidades académicas de la Universidad Católica de Córdoba a partir de la Resolución Rectoral N° 2775/20 en Octubre de 2020, con el objetivo de:

- Brindar espacios y herramientas que faciliten el acceso y la permanencia en la institución.
- Favorecer la formación de grupos de estudio coordinados por tutores.
- Generar espacios para el encuentro, trabajo en grupo y discusión de ideas.

El programa cuenta con un proceso de selección y designación de tutores en las distintas áreas de conocimiento, quienes son dados de alta por parte de los coordinadores para ejercer su función de tutor. Cada unidad académica tiene su propio proceso y requisitos para la selección y designación.

Tras analizar la propuesta del programa y nuestra experiencia como tutores, destacamos varias áreas problemáticas que sugieren la posibilidad de mejora:

- La estandarización y unificación de un proceso de inscripción.
- La implementación de un sistema de feedback real que permita opiniones tanto de tutores como de alumnos.
- La dispersión de la información sobre los horarios de los tutores y la falta de una plataforma centralizada para consultarla.

Surge así la propuesta de mejorar la implementación del programa TAIE, tanto por la importancia que tienen las tutorías en la formación de estudiantes como por nuestra visión de mejorar un sistema del cual hemos participado a través de los conocimientos obtenidos en la carrera.

Objetivos

Nuestra propuesta de mejora al programa de Tutorías de Acompañamiento Integral a Estudiantes (TAIE) implica el desarrollo y despliegue de tres módulos de software conectados entre sí. Una aplicación móvil llamada “Turnero TAIE”, un tablero web y una base de datos. Este despliegue busca cumplimentar con esta serie de objetivos:

Objetivos Generales

- Mejorar la eficiencia y funcionamiento del programa de Tutorías de Acompañamiento Integral a Estudiantes (TAIE) mediante la implementación de tecnología.
- Optimizar los procesos internos de inscripción, selección, contacto y seguimiento de las tutorías. Implementar un sistema centralizado para estos procesos.
- Proporcionar un flujo de información completo y actualizado sobre las instancias de tutorías próximas, actuales y pasadas.

Objetivos Específicos

- Desarrollar una aplicación móvil para facilitar a los tutores la gestión de sus horarios disponibles y sus respectivas asignaturas.
- Implementar una base de datos que permita almacenar y recuperar información relevante sobre los tutores, alumnos y áreas de conocimiento.
- Crear un tablero web interactivo para que los coordinadores puedan acceder fácilmente a la información de los tutores en proceso de inscripción, y puedan recuperar feedback sobre el programa a lo largo del año.
- Proporcionar una plataforma para que los tutores y los alumnos puedan coordinar sus sesiones de tutorías de manera más efectiva.

Marco Teórico

Tutoría universitaria

En la sociedad contemporánea, las universidades se han consolidado como entidades fundamentales para la formación y educación académica de los individuos. Sin embargo, persisten en ellas modelos de enseñanza tradicionales que han experimentado cambios mínimos desde su creación.

Los rápidos y constantes avances tecnológicos generan significativas transformaciones culturales y de comportamiento. Es necesario entonces, como institución, cuestionar y replantear los modelos de enseñanza tradicionales. Este proceso implica la adopción de técnicas y herramientas que trascienden las clases convencionales, con el objetivo de proporcionar un aprendizaje efectivo e integral. Este enfoque no solo abarca conocimientos técnicos y académicos, sino que también busca brindar a los alumnos orientación y un sentido de integración en la comunidad académica.

Una gran herramienta que aborda estas dificultades es el **"knowledge sharing"** o compartir conocimiento, aplicado al modelo de tutorías entre pares. Se trata de un enfoque educativo en el que, a través de la colaboración y el intercambio de ideas entre los estudiantes, se fomenta un aprendizaje interactivo y participativo. En este proceso, la experiencia de un estudiante avanzado facilita la comprensión de los temas estudiados, promoviendo un sentido de comunidad que aporta beneficios tanto académicos como sociales a los estudiantes que participan de estos encuentros. Surge entonces el concepto de **tutoría universitaria**, en donde contemplamos a los tutores como alumnos avanzados que aportan información y orientación a los estudiantes.

La tutoría universitaria, según la definición de Echeverría (1997), se concibe como “La actividad del profesor tutor encaminada a propiciar un proceso madurativo permanente, a través del cual el estudiante universitario logre obtener y procesar información correcta sobre sí mismo y su entorno, dentro de planteamientos intencionales de toma de decisiones razonadas: integrar la constelación de factores que configuran su trayectoria vital; afianzar su autoconcepto a través de experiencias

vitales en general y laborales en particular; desplegar las habilidades y actitudes precisas, para lograr integrar el trabajo dentro de un proyecto de vida global.”

La tutoría universitaria, según García Nieto, N. (2008), contribuye a la formación e integración del estudiante en la comunidad académica en cinco ámbitos :

- **Académico**: Se refiere a aspectos de la vida universitaria, como planes de estudios, elección de asignaturas, selección de vías de especialización, grados universitarios, postgrados, másteres y cursos de especialización.

- **Profesional**: Implica asesoramiento y ayuda en la inserción socio-laboral, facilitación del tránsito desde la universidad hacia la vida activa, estudio de ofertas y demandas de empleo, y prácticas en empresas.

- **Personal**: Relacionada con problemas personales, familiares, psicológicos, emocionales y afectivos que pueden afectar directa o indirectamente al aprendizaje del estudiante y a su desarrollo personal y profesional.

- **Social**: Destinada a cuestiones como información sobre ayudas y servicios universitarios disponibles, consecución de becas, estancias en el extranjero e intercambio de estudiantes, y movilidad estudiantil.

- **Administrativa**: Referida a temas como información sobre requisitos administrativos, matriculación, convalidaciones y biblioteca.

Situación en la Universidad

A partir de 2014, se identificaron diversas dificultades en el ámbito educativo de la Universidad Católica de Córdoba, según la evaluación realizada por la Secretaría Técnica, la Coordinación y la Asesoría Pastoral de la Facultad. Estas preocupaciones incluían la deserción temprana de los estudiantes, quienes abandonan las carreras en los primeros años de cursado, la reiterada dificultad y recursado de asignaturas específicas, especialmente aquellas consideradas troncales, y la falta de adaptación e integración al sistema universitario durante el ciclo básico.

Como respuesta a estas problemáticas, comenzaron a debatirse y gestarse distintas soluciones posibles para mejorar el bienestar de la comunidad interna en la UCC, enfocándose en particular en la estancia y no deserción de los alumnos del ciclo básico.

Programa de Tutorías de Pares (PTP)

En 2015 se implementa en la facultad de Ingeniería el Programa de Tutorías de Pares (PTP), un proyecto creado por Judith Disderi en conjunto con Gustavo Chiodi, Ariel Uema, Mabel Pautasso y Matías Aller. Este programa tuvo como objetivo mejorar la inserción y permanencia en las carreras de Ingeniería en los estudiantes del ciclo básico. Se destaca que no solo se busca proporcionar ayuda con los temas dictados en las asignaturas troncales, sino también enseñar cómo manejarse en el ámbito universitario. Propone una colaboración entre alumnos que brindan soporte académico en las diversas asignaturas y brindan información sobre actividades extraacadémicas, becas, o técnicas de estudio que mejoren el rendimiento académico.

Propuesta PTP

La instancia inicial del Programa de Tutorías de Pares establece los siguientes lineamientos:

En cuanto al proceso de selección para aplicar al rol de tutor, es necesario enviar un correo electrónico a ingsec@uccor.edu.ar (Secretaría de Grado). Esta inscripción se encontraba abierta durante todo el año, por lo que podían haber cambios en los tutores disponibles en cualquier momento.

No se especifica qué criterios se tendrán en cuenta para la selección final de tutores que hayan aplicado, y se elige un único tutor por carrera (un (1) estudiante de Ing. en Sistemas, un (1) estudiante de Ing. Mecánica, etc.).

Para encontrar algún turno o conocer la disponibilidad de los tutores, era necesario consultar las redes sociales de la facultad o los 'tableros' físicos en la propia facultad, haciendo referencia a documentos disponibles en los tableros de aviso de la Facultad de Ingeniería, y contactar a los tutores por correo electrónico o mensajería movil (WhatsApp) de manera independiente.

No se contaba con ningún tipo de registro ni control sobre las tutorías dictadas, por lo que el feedback real era nulo. Solo se contaba con el recuerdo anecdótico de los tutores.

Por último, no existía ningún tipo de motivación o recompensa que invitara y alentara a los estudiantes a participar en este programa como tutor, como podría ser la obtención de puntos RSU.

Estaba claro que esta primera iteración tenía muchas cosas por corregir, pero con la voluntad de sus creadores y los primeros participantes, pudieron concluir con éxito el primer año del programa.

El Programa de Tutorías de Pares continuó evolucionando a lo largo de los años, a través del feedback de los tutores, tutorandos y la disposición de los creadores del programa.

Algunas de las mejoras a destacar son:

- Mayor cantidad de estudiantes tutores.
- Asignación de tutores por grupos de materias y no por carreras.
- Criterios de selección claros.
- Mayor difusión sobre los horarios de los tutores y materias disponibles a ser dictadas en clase.
- Obtención de puntos de Responsabilidad Social Universitaria a los alumnos participantes en carácter de tutor.

Estas notables mejoras provocaron una mayor difusión del programa, atrayendo nuevos tutores ano tras anio. Este crecimiento no solo se dio en la Facultad de Ingeniería, sino también en todas aquellas con un programa similar al de Tutorías de Pares.

Con el objetivo de crear un programa estandarizado para todas las unidades académicas, por iniciativa de la Secretaría Académica, en Octubre de 2020 se crea el programa de Tutorías de Acompañamiento Integral a Estudiantes (TAIE), iniciativa que obtuvo la autorización mediante la Resolución Rectoral Nº 2775.

Esta resolución intenta agrupar los distintos modelos practicados en las unidades académicas en uno solo, aprovechando los aspectos positivos de cada uno de los programas individuales.

Tutorias de Acompañamiento Integral a Estudiantes (TAIE)

La resolución nos introduce a los problemas identificados en el contexto académico, y como distintas iniciativas surgieron en las unidades académicas para abordar las dificultades de inserción universitaria y desempeño académico que los estudiantes presentaban durante los primeros años de la carrera.

Se plantean distintas definiciones que justifican la utilización de esta herramienta de acompañamiento entre estudiantes para un mejor desempeño e integración:

“La Tutoría puede ser considerada, en sus diversas manifestaciones, como una estrategia que, articulada con otras, contribuye a la integración de los estudiantes al grupo de pares y a la Universidad, al sostén de la trayectoria formativa y al mejoramiento del aprendizaje académico” (RESOLUCIÓN RECTORAL N° 2775/20)

“Como dispositivo pedagógico, la Tutoría abarca distintas formas de acompañamiento y seguimiento de la trayectoria personal de aprendizaje de los estudiantes y consiste en la delimitación conjunta de apoyos académicos y apoyos personales específicos. Uno de sus principales propósitos es generar mejores condiciones para la retención del estudiantado, la enseñanza y el aprendizaje

mediante intervenciones en el plano de las dimensiones vincular y académica de la vida universitaria" (RESOLUCIÓN RECTORAL N° 2775/20)

Intervención "de carácter intencionado, que consiste en el acompañamiento cercano al estudiante, sistemático y permanente, para apoyarlo y facilitarle el proceso de construcción de aprendizajes de diverso tipo: cognitivos, afectivos, socioculturales y existenciales" (Narro Robles & Arredondo Galván, 2013, p. 138).

Una vez fundamentados estos principios, nos centramos en el diseño de la nueva propuesta.

Propuesta TAIE

La siguiente es una extracción de la propuesta formalizada en la Resolución Rectoral N° 2775/20 del programa de Tutorías de Acompañamiento Integral a Estudiantes (TAIE), con el objetivo de clarificar el funcionamiento y objetivos del modelo actual.

I . Objetivos del TAIE

- *Brindar espacios y herramientas que contribuyan al acceso y la permanencia en la institución, buscando favorecer la formación de grupos de estudio coordinados por tutores (estudiantes avanzados o docentes), quienes guían el trabajo con diferentes materiales, libros, apuntes, etc.*

- *Generar espacios complementarios (físicos, digitales e institucionales) para el encuentro, trabajo en grupo y discusión de ideas que posibiliten la apropiación de contenidos curriculares por parte de los estudiantes y contribuyan a la consolidación de una identidad institucional.*

II. Acerca de los/las tutores/as

II. 1- El proceso de selección y designación de tutores constará de las siguientes instancias:

- *Apertura de la Convocatoria a cargo de la SPyRSU y SPU.*

- *Selección de asignaturas que requieren acompañamiento tutorial y/o desarrollo de habilidades y/o competencias para fortalecer la trayectoria académica, difusión de convocatoria, inscripción de postulantes a cargo de la Unidad Académica*
- *Selección de tutores y notificación de resultados a cargo de la SPyRSU y SPU.*

II. 2- Requisitos para aspirar a desempeñarse como TUTOR:

- *Estudiantes de grado:*

1. *Ser estudiante de segundo año en adelante de las carreras de las unidades académicas correspondientes.*
2. *Tener aprobadas las asignaturas en las cuales se desempeñe como tutor.*
3. *Tener un genuino interés y deseo de acompañar las trayectorias estudiantiles en el proceso de integración académica e institucional a la vida universitaria*
4. *Mostrar una clara adhesión a nuestro "Nuestro modo universitario de proceder: Código de convivencia de la UCC" manifestado por aval del Decano/a. **

- *Docentes de la UCC:*

1. *Ser preferentemente egresado/as de grado o posgrado de la Universidad.*
2. *Tener un genuino interés y deseo de acompañar las trayectorias estudiantiles en el proceso de integración académica e institucional a la vida universitaria.*
3. *Tener solidez académica en los contenidos de las asignaturas en las cuales se va a desempeñar como tutor.*
4. *Mostrar una clara adhesión a nuestro "Nuestro modo universitario de proceder: Código de convivencia de la UCC" manifestado por aval del/de la Decano/a.*

II. 3- Funciones del/de la TUTOR/A

- *Atender dudas académicas de estudiantes, particularmente aquellas que involucran conceptos básicos propios de las asignaturas del ciclo básico (primer y segundo año)*
- *Capacitarse para el desempeño de sus tareas.*

- *Orientar al/a la estudiante en estrategias de estudio.*
- *Identificar dificultades y desarrollar acciones pertinentes para resolverlas (derivación al Servicio de Orientación y Aprendizaje (SOA), Comisión de Bienestar, dialogar con docentes, entre otras)*
- *Estimular a los/las alumnos/as a estudiar de manera autónoma.*

III. Estudiantes que participan de las tutorías

- *El Programa TAIE enfoca sus actividades principalmente en la orientación de alumnos/as del ciclo básico (primer y segundo año de las carreras).*
- *Las consultas a los/las tutores/as por parte de los/las alumnos/as es voluntaria, por propia iniciativa o por sugerencia de otros actores institucionales.*

IV. Áreas de articulación

- *Área Pedagógica (SPU): orientar y capacitar a docentes y estudiantes en lo referente al rol de tutor.*
- *Área Académica (Cada UA): seleccionar las asignaturas, contenidos sobre los cuales se centrarán las tutorías año a año, en cada UA, organizar el cronograma de tutorías en la propia unidad académica, promover la participación docente y estudiantil como tutores, velar por el desarrollo del programa en la UA.*
- *Área de RSU: apertura y cierre de la convocatoria, acreditación académica de docentes y estudiantes participantes como tutores, organización de las instancias de trabajo conjunto con las UA (selección de tutores y evaluación final).*

V. Circuito procedimiento

Momento del ciclo	Actividad	Área responsable
1	Selección de asignaturas que	UA

	requieren acompañamiento tutorial	
2	**Diseño de propuesta y cronograma de formación pedagógica y técnica para tutores/as**	**SPU**
3	**Preinscripción de estudiantes tutorados**	**UA**
4	**Apertura y difusión de convocatoria de TUTORES/AS docentes y estudiantes**	**SPyRSU**
5	**Selección de tutores/as**	**SPU+UA+SPyRSU**
6	**Comunicación de resultados a tutores/as**	**UA**
7	**Capacitación tutores/as**	**SPU**
8	**Inicio del TAIE**	**UA**
9	**Seguimiento de actividades**	**UA**
10	**Instrumento de evaluación final de resultados**	**SPyRSU+SPU**
11	**Aplicación de instrumento de evaluación**	**UA**
12	**Evaluación conjunta**	**UA-SPU-SPyRSU**

Análisis de TAIE

En esta propuesta formalizada, destacamos los siguientes elementos:

- Requisitos claros para los aspirantes a tutores: Se establecen requisitos comprensibles para aquellos que deseen desempeñarse como tutores. Se deja en claro que la selección considerará el interés y deseo genuino del tutor, así como la aprobación de las asignaturas en las que llevará a cabo su labor.

- Involucramiento del área de Responsabilidad Social Universitaria (RSU): El área de RSU ahora participa activamente, encargándose de la apertura y cierre de la convocatoria. Además, acredita a los estudiantes participantes como tutores, motivándolos a involucrarse de manera activa en el programa.

- Seguimiento y colaboración universitaria: En el análisis del procedimiento, se destaca un seguimiento de las actividades por parte de cada Unidad Académica. Además, se resalta la colaboración del área pedagógica para capacitar a los tutores en sus labores, evidenciando un trabajo conjunto y una mayor integración entre distintas áreas académicas.

La formalización del programa TAIE presenta una gran cantidad de mejoras, pero esta resolución no detalla los pasos intermedios tomados por cada unidad académica para coordinar la reunión final entre alumno y tutor, como la búsqueda de información sobre los horarios disponibles de los tutores o las asignaturas que dictan.

Para abordar este aspecto, proponemos completar este contexto faltante con una descripción de nuestra experiencia como tutores pares participantes del programa TAIE en la Facultad de Ingeniería. Esta narrativa ayudará a llenar los vacíos en la información y ofrecerá una perspectiva valiosa sobre el funcionamiento práctico del programa desde la experiencia directa de los tutores.

Experiencia propia como tutores

Ambos autores nos inscribimos como tutores pares a principios de 2020, participando además ocasionalmente como alumnos, hasta 2023. Debido a esto, vivimos la transición del modelo de Programa de Tutores Pares (PTP) al nuevo programa universitario de Tutorías de Acompañamiento Integral a Estudiantes (TAIE) desde ambas perspectivas. Describimos a continuación aquellas secciones del proceso que se encuentran sin mencionar en la propuesta aprobada en la resolución rectoral.

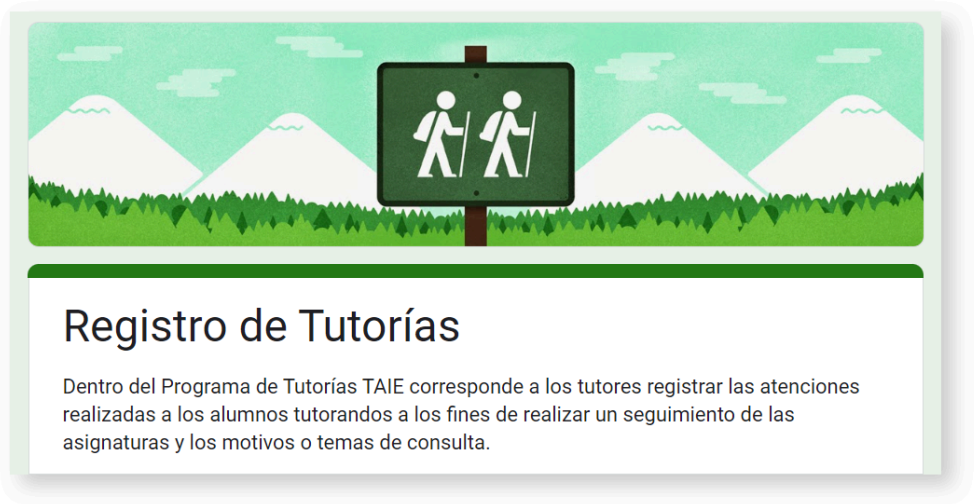
- Proceso de inscripción: en lugar de enviar un correo electrónico a la Secretaría Académica, los tutores interesados en participar en el programa ahora debían completar un formulario de Google llamado "Ficha de Inscripción Programa TAIE". Este formulario solicitaba información como la clave UCC, carrera y el grupo de materias en el que el tutor estaba interesado. Cada Unidad Académica gestionaba su propio formulario, agrupando por materias o "áreas de conocimiento". Sin embargo, la difusión y disponibilidad del enlace del formulario a menudo no eran claras ni fáciles de encontrar.

- Proceso de selección: debido al aumento en la participación, motivado principalmente por la asignación de puntos RSU al participar como tutor, llevó al ajuste de los criterios de selección por parte del coordinador. A partir de este ajuste, se consideraron nuevos factores como la cantidad de puntos RSU faltantes, el número de veces que el alumno había participado como tutor en instancias anteriores del programa, la cantidad de estudiantes de esa carrera que ya habían sido admitidos como tutores, y el rendimiento académico en las materias o áreas de conocimiento seleccionadas por el estudiante para dictar clases.

- Contacto y coordinación de tutorías: se hizo hincapié en la difusión del contacto de los tutores, publicando en todos los medios de comunicación universitarios la lista de tutores junto con su área de conocimiento asociada, el horario disponible para contactar cada tutor y en el area de contacto encontrabamos el numero de telefono y el correo electronico personal. Los alumnos tenían la responsabilidad de tomar la iniciativa y contactar a los tutores directamente a través de correo electrónico o WhatsApp.

- Feedback y registro de tutorías: para realizar un seguimiento sobre las atenciones y clases realizadas a los alumnos tutorandos, se implementó el

formulario "Registro de Tutorías". Este formulario, gestionado a través de Google, debía ser completado por cada tutor después de cada atención a uno o más alumnos. En el mismo, se debe proveer manualmente el nombre y apellido del alumno que recibió la clase, acompañado de su clave de alumno. Además debe seleccionar a qué área de conocimiento pertenece la consulta, agregar detalles sobre el tema de consulta y en caso que sea necesario agregar observaciones varias.



Formulario “Registro de Tutorías”

Si bien estas nuevas metodologías y consideraciones superan al Programa de Tutores Pares (PTP), encontramos varios desafíos que necesitan ser abordados para optimizar el Programa de Tutorías de Acompañamiento Integral a Estudiantes (TAIE). A raíz de esto surge el análisis formalizado de la problemática a tratar.

Problemática en el Programa TAIE

La problemática identificada en el programa de tutorías TAIE abarca diversos aspectos que afectan su eficiencia y funcionamiento. A continuación, se destacan algunas de estas áreas de mejora.

Inscripción al programa

- Si bien el programa se encuentra unificado en toda la universidad, cada unidad académica realiza por su cuenta el formulario de inscripción en línea. Se pierde la uniformidad en formularios, solicitudes y periodos de inscripción, provocando que existan diferentes procedimientos y requisitos según la facultad a la que pertenezcan los estudiantes. Esta falta de consistencia puede generar confusión entre los alumnos y no permite la existencia de un consenso general en los métodos de inscripción.

- Las instancias de inscripción dependen en gran medida de procesos manuales, tanto en su creación, como en la correcta difusión y distribución del enlace de acceso en los canales universitarios apropiados. La falta de un sistema centralizado y uniforme genera confusiones y obstáculos para los estudiantes interesados en participar como tutores o tutorandos.

Sistema de feedback y retroalimentación

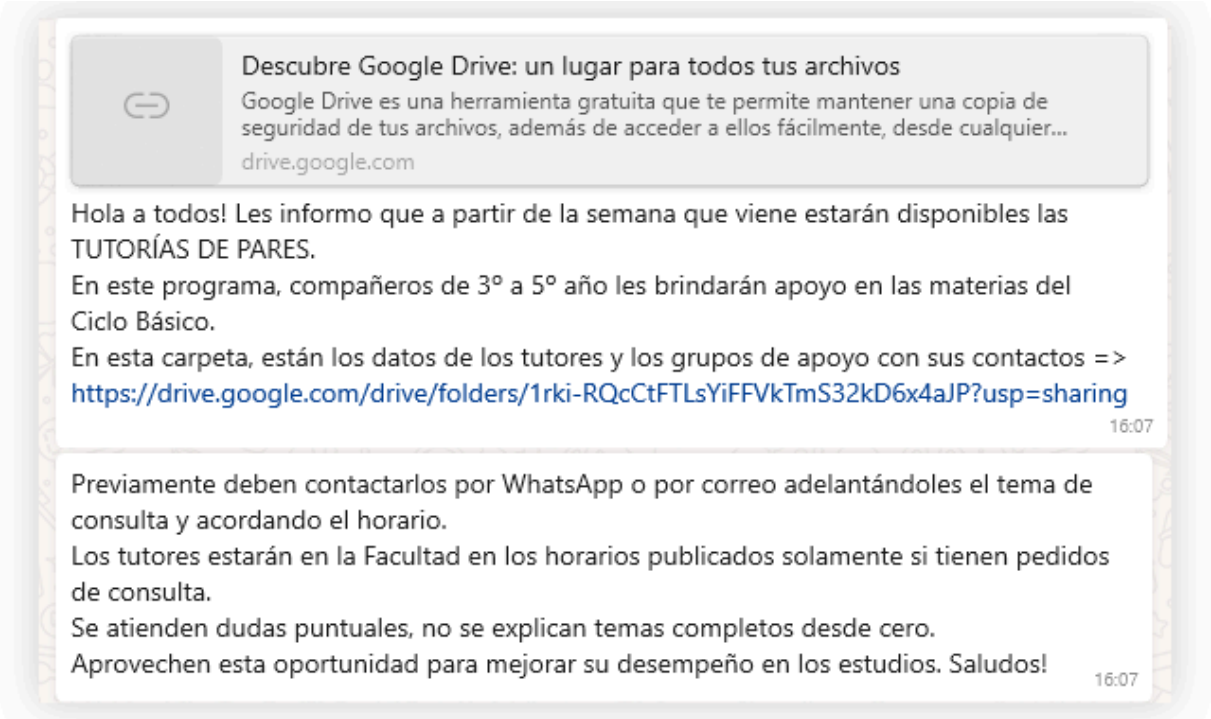
- La forma actual en la cual se recoge el feedback del programa presenta varias limitaciones. Los tutores deben recordar manualmente de qué forma se dictó la clase y cuáles fueron las principales dudas y observaciones en la asignatura revisada. La falta de un límite para enviar el feedback resulta en la pérdida de información con el tiempo, el tutor puede olvidar cargar la instancia o detalles de la misma.

- El feedback del alumno sobre la tutoría no está contemplado, limitando la evaluación del proceso únicamente a la perspectiva del tutor.

- Toda la información del feedback queda en formularios de Google aislados, sin una forma de obtener o consultar datos de manera centralizada. La falta de estadísticas ordenadas impide evaluar el rendimiento de las tutorías y tomar decisiones fundamentadas en datos concretos.

Contacto

- Existe cierta fricción en el contacto directo entre el tutor y el alumno. El programa TAIE busca integrar a la comunidad universitaria a aquellos alumnos que estén teniendo dificultades en hacerlo. El hecho de que estos alumnos tengan que contactar directamente a los tutores por iniciativa propia puede limitar el alcance y la participación de los alumnos tutorandos en el programa.



Mensaje enviado por WhatsApp por parte de la Secretaría de Grado.

- La dispersión del número de teléfono del tutor por todos los canales de difusión de la unidad académica puede generar vulnerabilidad y exposición innecesaria para los tutores.

TUTORES PARES 2022 - FACULTAD DE INGENIERIA UCC

Carrera	Apellido	Nombre	Teléfono	Mail	Año
Ingenieria Computación		Nicolas	35 9	1912145@ucc.edu.ar	4
		Leonel	29 7	1900348@ucc.edu.ar	4
Ingeniería Electrónica		Lautaro	35 1	2000718@ucc.edu.ar	3
Ingeniería Industrial		Paul	35 5	2015203@ucc.edu.ar	5
		Matias Julian	29 0	1714715@ucc.edu.ar	5
		Agustin	35 9	1913838@ucc.edu.ar	5
		Victoria	35 0	1811982@ucc.edu.ar	5
		Agustina	35 5	1814622@ucc.edu.ar	5
		Natalia	35 9	2008315@ucc.edu.ar	3
		Enrique	35 2	1504454@ucc.edu.ar	5
Ingeniería Mecánica		Leonel David	35 8	1900116@ucc.edu.ar	4
Ingeniería Civil		Conrado Jure	35 4	1704443@ucc.edu.ar	5
		Juan Ignacio	35 1	1701290@ucc.edu.ar	5
Ingeniería Sistemas		Ramiro	38 6	1906675@ucc.edu.ar	4
		Santiago Agustin	35 0	2000049@ucc.edu.ar	3
		Valentino	35 8	2113503@ucc.edu.ar	3
		Juan	35 0	1902260@ucc.edu.ar	4
Lic. Bioinformática		Martina	35 1	1923373@ucc.edu.ar	4

Tablero físico/virtual de Tutores con numero de telefono.

Horarios

- La existencia de distintos documentos con los horarios de los tutores, tanto físicos como digitales, y la necesidad de actualización manual por parte de los coordinadores de cada unidad académica en caso de un cambio necesario, dificultan la identificación de una fuente de información centralizada y actualizada. Esto restringe la flexibilidad de los tutores para cambiar horarios, materias asignadas o darse de baja del programa, y puede dar lugar a información desactualizada en avisos físicos en la facultad.

GRUPOS DE APOYO de TUTORES PARES 2022 - FACULTAD DE INGENIERIA UCC

TAIE	APOYO TRANSVERSAL		APOYO ACADEMICO							
GRUPOS	Acompañamiento motivacional	Inducción de alumnos ingresos tardíos / pase	MATEMATICA (AMI-II-III, AyG, AN, EyP)	FISICA-QUIMICA (Física I-II-III, Mecánica, Química General)	PROGRAMACION (Fund.Prog, Prog, Progl, Herr.Inf.)	DIBUJO TECNICO (SR, SRG, SRA)	BIOINFORMATICA (QOI, Met.Inv. Biología, Intr. Bioinf, Bioinf.I)	CIVIL (OCyA, Topol, AE1)	INDUSTRIAL (SI, IO1)	TICS (LógicayMD, Prog.I-II, Lab.Comp.I-II, Med.Eléctricas)
ATLUTMONROESS	Santiago Riveros Salomón	Amuchastegui Enrique	Santiago Riveros Salomon	Lujan Lautaro	Ghilino Ramiro	Ruatta C. Leonel	Ceballos Martina	Srur, Juan Ignacio	Amuchastegui	Santiago Riveros Salomón
	Lujan Lautaro	Rizzoli, Paul	Lujan Lautaro	Sole Leonel	Sole Leonel	Amuchastegui		Santacroce, Conrado	Rizzoli, Paul	Lujan Lautaro
	Gaggio Valentino	Gaggio Valentino	Vergara Natalia	Gross Agustin	Nicolas Leszezyñski	Lopez Romero			Vergara Natalia	Ghilino Ramiro
	Gross Agustin	Gross Agustin	Ficca Matias	Ruatta C. Leonel	Gaggio Valentino	Ficca Matias			Gross Agustin	Sole Leonel
	Amuchastegui Enrique	López Romero							Lopez Romero	Nicolas Leszezyñski
	Rizzoli, Paul									Gaggio Valentino
Link al grupo de wsp	Contactar personalmente a cada tutor	Contactar personalmente a cada tutor	https://chat.whatsapp.com/J40PvgLTNcBHmsMsnwE2aH	https://chat.whatsapp.com/Gn3Zg9ZqlfHFecaT4zOnFY	https://chat.whatsapp.com/LXncnqiyJCT5IHFTPdGEdt	https://chat.whatsapp.com/HV4F0Sep2uuInPF9LzDTqE	https://chat.whatsapp.com/DDDYGMQfy3ALSeQEyvwT2f	https://chat.whatsapp.com/KLe8LfUZ64250uOSTIwhq	https://chat.whatsapp.com/JD5AOGE25Mz7G73iEgDWgk	https://chat.whatsapp.com/LThSIALBYoiIc9bCkIpPej
HORARIO DE CONSULTA 1 (dia-hora)	Contactar personalmente a cada tutor	Contactar personalmente a cada tutor	Jueves de 17hs a 18hs Campus (Santiago) Martes de 14hs a 15hs Campus (Lautaro)	Jueves de 13 a 15 hs Campus (Leonel S)	Jueves de 12 a 15 hs Campus	Martes de 15 a 16hs Campus (Leonel)	Miercoles de 12hs a 13hs en el Campus	Contactar tutor	Contactar Tutor	Jueves de 12 a 15 hs Campus
HORARIO DE CONSULTA 2 (dia-hora)	Miercoles 15 a 16 hs (Enrique)	Miercoles 15 a 16 hs (Enrique)	Viernes de 16 a 17hs Campus (Natalia)	Viernes de 13 a 15hs campus (Lautaro)		Miercoles 15 a 16 hs (Enrique)			Miercoles 15 a 16 hs (Enrique)	Viernes 14hs a 15.45 hs Campus

Tablero virtual de horarios en diferentes grupos de apoyo

Debido al rol fundamental que las tutorías desempeñan dentro de la comunidad académica como revisamos anteriormente, y los beneficios que traen estas prácticas a los alumnos de ciclo básico, surge la necesidad de revisar y proponer mejoras para el programa de "Tutorías de Acompañamiento Integral a Estudiantes" (TAIE).

Soluciones similares

Es importante realizar un repaso sobre soluciones existentes que abordan los problemas identificados en este trabajo previo al desarrollo del mismo, entendiendo el modelado de plataformas de gestión y calendarización de encuentros. A continuación, se presenta una breve mención de una de estas soluciones:

Calendly

Calendly es una herramienta de programación en línea que facilita la coordinación de reuniones y eventos al permitir que los usuarios compartan sus disponibilidades y permitan que otros programen citas en esos horarios. Se encuentra enfocado principalmente a la coordinación de reuniones en empresas de manera virtual.

La aplicación permite integraciones con el calendario nativo de cada usuario, además de ofrecer múltiples módulos de integración con plataformas web reconocidas como LinkedIn, Zoom y varios CRMs.

Su principal desventaja radica en su enfoque empresarial, vinculado a las ventas y la coordinación de llamadas con potenciales clientes. Aunque existe un modelo educativo, pagar por este modelo sin la implementación del modelo empresarial podría no representar un buen retorno de inversión.

Skooli y plataformas de tutores online

Skooli es una plataforma de tutorías en donde es posible solicitar tutorías en línea instantáneas. A través de un motor de búsqueda, se puede acceder a todas las materias que se dictan en la plataforma en cualquier momento y lugar. Su objetivo es proporcionar tutorías individuales para estudiantes que buscan ayuda con sus estudios.

Esta y muchas otras plataformas de tutoría solo están disponibles en inglés y no proporcionan una solución para la coordinación de reuniones físicas ni un contacto entre alumnos, ya que solo se realiza con tutores disponibles en la página.

Plataforma de interacción

Los usuarios deben interactuar con el sistema a desarrollar. Existen distintas plataformas en las cuales desplegar nuestro sistema.

Aplicación móvil

Una aplicación móvil es un pequeño paquete de software diseñado para funcionar en dispositivos móviles, como smartphones o tablets. Este tipo de aplicaciones permite al usuario acceder a funcionalidades específicas desde sus dispositivos. Para usar este software, el usuario debe consultar una tienda de aplicaciones, como la Google Play Store en dispositivos con sistema operativo Android o la App Store en dispositivos con sistema operativo iOS, y descargar la aplicación. Pueden ser descargadas de forma gratuita o pueden incurrir un costo para el usuario.

Las aplicaciones móviles presentan una arquitectura cliente-servidor, donde el dispositivo móvil actúa como cliente y se comunica con un servidor para obtener o enviar datos. Entre sus principales ventajas encontramos su accesibilidad en cualquier momento y lugar, su capacidad para aprovechar las funcionalidades específicas de los dispositivos móviles, y su capacidad para ofrecer una experiencia de usuario intuitiva y adaptada a la pantalla táctil.

Aplicación web

Una aplicación web, también conocida como 'WebApp', es una aplicación accesible a través de un navegador web en cualquier dispositivo con conexión a internet. Su arquitectura está basada en tecnologías web como HTML, CSS y JavaScript, y puede estar respaldada por un servidor para el procesamiento de datos.

Las principales ventajas de una aplicación web son su accesibilidad multiplataforma, ya que no requiere instalación y es compatible con cualquier dispositivo con un navegador web actualizado, y su facilidad de actualización, ya que los cambios en el software se aplican instantáneamente para todos los usuarios.

Aplicación de escritorio

Una aplicación de escritorio es un software diseñado para ser ejecutado en una computadora personal o portátil. Se instala localmente en el dispositivo del usuario y puede acceder a recursos locales, como archivos y periféricos. Ejemplos de sistemas operativos que permiten la instalación de aplicaciones son Windows, Linux y macOS.

Las principales ventajas de una aplicación de escritorio incluyen su capacidad para funcionar sin conexión a internet, su rendimiento optimizado para el hardware local y su capacidad para acceder a recursos locales sin depender de una conexión de red.

Desarrollo en aplicación móvil

Las siguientes son alternativas para la plataforma de desarrollo móvil propuesta para nuestro sistema. Muchas de ellas cuentan con el respaldo de grandes empresas, dado que el mundo de las aplicaciones móviles es uno de los ecosistemas más amplios y utilizados en el ámbito del software en la actualidad.

Entre las variables más importantes se encuentran la facilidad de implementación o curva de aprendizaje en base a lo conocido por los autores, así como también la reutilización del código para ambas plataformas, iOS y Android.

Las plataformas ideales serán aquellas que permiten escribir el código una única vez, pero permitir que corra en varios dispositivos. Es decir, aquellas que brinden un soporte multiplataforma.

Plataformas nativas

Estas son plataformas especializadas para el desarrollo nativo de un solo sistema operativo. En el caso de Kotlin, está orientado al desarrollo en Android, mientras que Swift está orientado al desarrollo nativo en iOS.

Investigamos plataformas nativas para conocer en qué se diferencian unas de otras, y así evaluar la factibilidad de desarrollar la aplicación desde ambos lenguajes en paralelo.

Kotlin

Kotlin es un lenguaje de programación de propósito general y estáticamente tipado que se ejecuta en la máquina virtual de Java (JVM) y también puede compilarse a JavaScript o código nativo. Más del 50% de los desarrolladores de Android profesionales utilizan Kotlin como su lenguaje primario, y más del 70% de ellos indican que Kotlin los hace más productivos. A la hora de utilizarlo en el desarrollo de

Android, los equipos que adoptan Kotlin lo hacen por un aumento de la productividad, ya que el código en Kotlin es conciso, y por la amplia comunidad de desarrolladores. El 95% de las 1000 mejores aplicaciones de Android contienen código de Kotlin, y el desarrollo de Android es Kotlin-first. También fue adoptado por Google, donde más de 70 aplicaciones utilizan Kotlin.

Contiene protección contra las excepciones de punto nulo (null-safe), la causa número 1 de accidentes en Google Play.

Una de las ventajas clave de Kotlin es su interoperabilidad con Java, lo que permite a los desarrolladores utilizar bibliotecas y herramientas existentes de Java en proyectos Kotlin y viceversa. Esto facilita la adopción gradual de Kotlin en proyectos existentes y permite a los desarrolladores aprovechar las fortalezas de ambos lenguajes.

Swift

Swift es un lenguaje de programación desarrollado por Apple, presentado por primera vez en 2014. Diseñado específicamente para el desarrollo de aplicaciones en las plataformas de Apple, Swift se ha convertido en la principal opción para desarrolladores que crean aplicaciones para iOS, macOS y el resto de dispositivos del ecosistema. Swift se destaca por su sintaxis clara y concisa, que hace que el código sea más legible y mantenible en comparación con sus lenguajes predecesores: C, C++ y Objective-C. Además, Swift incluye características de bajo nivel como tipos, control de flujo y operadores, así como funcionalidades orientadas a objetos como clases y genéricos.

Al mantener soporte activo por parte de Apple, Swift es la primera opción a la hora de desarrollar aplicaciones nativas en iOS. Presenta un perfil enfocado en la velocidad, además de utilizar tecnología de compilación LLVM de alto rendimiento permitiendo sacar lo mejor del hardware del dispositivo.

Plataformas multiplataforma

Las siguientes son plataformas de desarrollo que permiten escribir una sola fuente de código que se compila para los distintos objetivos en un paquete nativo sin importar que se construye desde un código fuente unificado. Suele reducir en grandes cantidades el desarrollo que podría suponer escribir una aplicación en dos lenguajes nativos distintos, pero se pierde acceso a funcionalidades o a velocidades que solo podrían ser alcanzables al escribir aplicaciones en entornos de desarrollo especializados para las plataformas objetivo.

React Native

Es un marco de trabajo de código abierto creado por Facebook en 2015 que permite a los usuarios utilizar JavaScript y React junto con capacidades nativas de la plataforma para construir aplicaciones móviles.

Ofrece una amplia biblioteca de componentes de interfaz de usuario, acelerando el tiempo de desarrollo. Además, permite el acceso a las funcionalidades nativas del dispositivo, como la cámara, acelerómetro, etc.

Una gran herramienta que presenta es la posibilidad de escribir código específico para cada una de las plataformas en las que se lanzará la aplicación. Esto permite optimizar las aplicaciones nativas separadas utilizando código nativo para cada una de ellas. Presenta la funcionalidad de “recarga rápida” durante el desarrollo, que permite aplicar cambios en la aplicación sin la necesidad de volver a compilarlas. Las posibles desventajas residen en nuestro conocimiento del entorno de React, que es nulo.

Xamarin

Es un marco de trabajo de código abierto para el desarrollo de aplicaciones móviles multiplataforma fundado en 2011, siendo uno de los más antiguos de los tres. Xamarin cuenta con una amplia colección de paquetes útiles para el desarrollo multiplataforma, con paquetes como Xamarin.Essentials para el manejo de API multiplataforma y Xamarin.Forms que ofrecen interfaces de usuario de tipo nativo tanto en Android como en iOS.

La actividad de desarrollo en Xamarin se encuentra estancada en relación a sus competidores, siendo su antigüedad uno de los principales motivos, además de la antigüedad las plataformas que soportan su existencia (.NET y C#). Posee una curva de aprendizaje mediana a elevada, aunque cuenta con tutoriales gratuitos brindados por Microsoft.

Flutter

Flutter es un marco de trabajo de desarrollo de aplicaciones móviles multiplataforma de código abierto creado por Google en 2017. Permite desarrollar interfaces de usuario desplegables en Android, iOS y Web al mismo tiempo. Posee un gran rango de customización ya que al ser creado por Google tiene un fácil acceso a las librerías de Material Design, que cuenta con asombrosos widgets que mejoran enormemente la experiencia de usuario y la velocidad de desarrollo.

Flutter es altamente eficiente debido a que su código Dart se compila por encima de la biblioteca C, lo que lo acerca al código nativo de cada una de las plataformas en las que se ejecuta. Dart es muy similar a lenguajes orientados a objetos con los que tenemos experiencia, como lo puede ser C + +, por lo que aplana la curva de aprendizaje.

Cuenta con *PubDev*, un repositorio oficial de paquetes de Dart y Flutter. Importar paquetes con módulos ya construidos para adaptarlos a nuestra solución presenta increíbles ventajas a la hora de desarrollar widgets y funcionalidades extensas y complicadas.

Una desventaja se presenta a la hora de utilizar otro backend como alternativa a Firebase, ya que gran parte de la documentación sólo se refiere a la conexión con el servicio de Google únicamente.

Desarrollo en plataforma web

El desarrollo web abarca el proceso de crear y mantener sitios web de manera eficiente y escalable, con herramientas que faciliten la creación de interfaces de usuario responsivas y atractivas.

En el desarrollo web, es fundamental comprender que todas las soluciones utilizan tecnologías fundamentales como JavaScript, HTML y CSS, lo que significa que la elección de la plataforma de desarrollo reside en el framework que mejor se adapte a nuestras necesidades.

Svelte

Svelte es un compilador de front-end de código abierto que se introdujo inicialmente en 2016 y ha ganado popularidad entre la comunidad de desarrolladores 1.

Lo más destacable de Svelte es su rendimiento y eficiencia, ya que no requiere nada adicional para construir un DOM virtual, lo que resulta en que la aplicación en general tome mucho menos tiempo para cargar. Además, Svelte permite la creación de HTML, CSS, TypeScript y los combina en JavaScript plano, lo cual simplifica el desarrollo y reduce la cantidad de código necesario .

Las principales ventajas al utilizar Svelte son su simplicidad y rapidez en el desarrollo, gracias a su sintaxis sencilla y a su enfoque de compilar el código a JavaScript puro durante la fase de construcción, en lugar de interpretarlo en tiempo de ejecución.

Hay que considerar también que utilizando Svelte, se obtiene un código más limpio y optimizado, pero esto puede llevar a una menor adopción y compatibilidad con algunas herramientas o bibliotecas de terceros, especialmente aquellas que dependen de un DOM virtual o de la manipulación del DOM en tiempo de ejecución .

React

React es una librería para construir interfaces de usuario desarrollada por Facebook, conocida por su eficacia en la creación de componentes reutilizables y su enfoque en el manejo del estado de la aplicación.

Utilizando React, podemos crear aplicaciones web dinámicas y altamente interactivas, aprovechando su gran comunidad y el amplio ecosistema de herramientas y bibliotecas disponibles .

Muchas aplicaciones utilizan React, incluyendo Facebook, Instagram y Airbnb, y su éxito radica en su flexibilidad y la capacidad de integrarse con otras tecnologías y bibliotecas .

Es posible además utilizar NextJS, una extensión de React que añade soporte para renderizado del lado del servidor y generación estática de sitios web, lo cual es particularmente útil para SEO y para mejorar el rendimiento de las páginas.

Angular

Angular es un framework de desarrollo web de código abierto que se basa en TypeScript y fue lanzado inicialmente en 2010 bajo el nombre de "AngularJS", y luego se reescribió completamente, lanzando la versión Angular 2 en 2016.

Provee un enfoque basado en componentes, lo que significa que las aplicaciones se construyen a partir de piezas reutilizables y autónomas. Los componentes permiten una estructura jerárquica y modular, facilitando la construcción y mantenimiento de aplicaciones complejas.

Es utilizado en empresas como Google, Microsoft, Deutsche Bank, Forbes y PayPal, lo que demuestra su fiabilidad y escalabilidad para proyectos grandes y complejos

Arquitectura de implementación

La arquitectura de implementación es la estructura conceptual que establece los modelos, planificación, organización y principios que se toman en cuenta a la hora de desarrollar el sistema de software. Existen distintos patrones de arquitecturas que proveen modelos de componentes, interacciones y comunicación entre estos componentes, resolviendo problemas recurrentes.

A continuación, desglosamos las principales arquitecturas de implementación:

Arquitectura cliente-servidor

La arquitectura cliente-servidor es un patrón de diseño donde un cliente solicita un servicio y un servidor proporciona dicho servicio. Esta arquitectura es útil para situaciones donde la gestión centralizada de datos y dispositivos de red es necesaria, aunque puede resultar costosa debido al mantenimiento del servidor (3) .

Sus características más importantes son que separa la responsabilidad de la gestión de datos y servicios, permitiendo así una mayor flexibilidad y escalabilidad en la distribución de cargas de trabajo. Este enfoque se utiliza en aplicaciones como bancos, intercambio de archivos, correo electrónico y el World Wide Web

Arquitectura de microservicios

La arquitectura de microservicios es un estilo de arquitectura que divide una aplicación en pequeños servicios independientes que pueden ser desarrollados, desplazados y escalados de manera individual. Esto permite una mayor agilidad y adaptabilidad a las necesidades cambiantes del negocio .

El uso de microservicios hace que sea más fácil adoptar nuevas tecnologías y promover el desarrollo políglota, ya que cada servicio puede estar escrito en un lenguaje de programación diferente . Lo más importante en nuestra aplicación si

usamos microservicios es la escalabilidad horizontal y la capacidad de desarrollar cada servicio con la tecnología más adecuada para su caso de uso específico .

Suele utilizarse en aplicaciones modernas que requieren alta disponibilidad, escalabilidad y adaptabilidad rápida a los cambios .

Arquitectura MVC (Modelo-vista-controlador)

La arquitectura MVC o Modelo Vista Controlador implica dividir una aplicación en tres componentes principales: el modelo, que representa la lógica de negocio; la vista, que muestra la información al usuario; y el controlador, que gestiona la interacción entre el modelo y la vista .

La ventaja de este modelo es que facilita la separación de preocupaciones y la reutilización de componentes, lo que mejora la mantenibilidad y escalabilidad de la aplicación . Sus características más importantes son la modularidad, la capacidad de probar componentes de forma aislada y la posibilidad de tener múltiples vistas para un mismo modelo .

Suele utilizarse en aplicaciones web y de escritorio donde la separación de la lógica de presentación y la lógica de negocio es crucial para el manejo de la complejidad.

Arquitectura monolítica

La arquitectura monolítica es un estilo de arquitectura donde toda la aplicación se ejecuta como un único proceso, utilizando una única base de datos y escalando verticalmente.

El enfoque monolítico implica que todas las funcionalidades de la aplicación están interconectadas y cualquier cambio en una parte del sistema puede afectar a todo el sistema. Sus características más importantes son la simplicidad en términos de desarrollo inicial, ya que no hay necesidad de coordinar entre múltiples servicios, y la posibilidad de realizar cambios grandes sin afectar la funcionalidad existente.

Suele utilizarse en aplicaciones que tienen requisitos simples o cuando se desea un lanzamiento rápido al mercado, aunque puede resultar menos escalable y más difícil de mantener a largo plazo

Backend

El backend es la parte del sistema encargada del procesamiento y almacenamiento de datos. En este aspecto, es crucial considerar elementos como la escalabilidad, el rendimiento y la mantenibilidad del sistema, al mismo tiempo que se garantiza una integración válida con la arquitectura definida.

Starlette

Starlette es un framework de alto rendimiento y ligero para construir APIs web y aplicaciones web asíncronas en Python. Este framework se centra en la eficiencia y facilidad de uso, permitiendo a los desarrolladores construir aplicaciones con un alto número de conexiones concurrentes de manera eficiente. Starlette es especialmente adecuado para aplicaciones que requieren manejo de conexiones asíncronas y operaciones de bajo costo.

El enfoque de Starlette es proporcionar un marco de trabajo mínimo y sin opiniones, lo que significa que no impone una estructura de proyecto o patrones de diseño específicos. Esto da a los desarrolladores la libertad de organizar su código de la manera que consideren más adecuada para su aplicación.

Starlette incluye un sistema de enrutamiento eficiente que permite definir rutas y manejar solicitudes HTTP de manera declarativa. El enrutamiento es flexible y se puede combinar con otras características como parámetros de ruta, captura de grupos y subrutas.

Además de su capacidad para manejar solicitudes HTTP, Starlette también ofrece soporte para middleware y manejadores de eventos. Esto permite a los desarrolladores extender el comportamiento de la aplicación de manera modular y reutilizable.

En cuanto a la seguridad, Starlette proporciona herramientas para prevenir ataques comunes como Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF). También incluye soporte para autenticación y autorización, permitiendo a los desarrolladores implementar controles de acceso a nivel de aplicación.

Starlette es compatible con una variedad de bases de datos y puede trabajar con ORM como SQLAlchemy o Tortoise-ORM para facilitar la interacción con la base de datos. El framework no impone una base de datos específica, lo que significa que los desarrolladores pueden elegir la solución que mejor se adapte a sus necesidades.

Starlette tiene una comunidad activa y un ecosistema en crecimiento de paquetes y plugins que los desarrolladores pueden utilizar para agregar funcionalidades adicionales a sus aplicaciones. Esto incluye herramientas para la validación de datos, manejo de sesiones, y más.

En resumen, Starlette es una opción atractiva para desarrolladores que buscan crear aplicaciones web y APIs eficientes y escalables en Python, ofreciendo un equilibrio entre simplicidad y potencia.

Django

Django es un framework de desarrollo web avanzado y de código abierto escrito en Python. Su diseño se centra en incrementar la eficiencia del desarrollador y promover la creación de código limpio y reutilizable, lo que lo ha posicionado como uno de los frameworks web más influyentes y sólidos en el ámbito del desarrollo.

Django sigue un patrón arquitectónico conocido como Modelo-Vista-Plantilla (MVT), aunque fuera del contexto de Django es denominado Modelo-Vista-Controlador (MVC), donde el modelo representa la estructura y lógica de los datos y entidades de la aplicación. La vista es responsable de la lógica de presentación y procesamiento de datos, interactuando con el modelo cuando sea necesario; la vista no solo utiliza la información almacenada en los modelos y la información de la petición, sino que también puede interactuar con el modelo para realizar operaciones adicionales, como la actualización de datos. Además, el procesamiento de la vista no siempre conduce a la renderización de una plantilla, ya que puede devolver diferentes tipos de respuestas, como redirecciones o respuestas JSON, dependiendo de la lógica de la aplicación. Finalmente, la plantilla se encarga de la presentación y renderización de la interfaz de usuario, permitiendo un control detallado sobre la salida generada y enviada al cliente. Esta estructura facilita la organización modular del código, separando la lógica de presentación de la lógica de negocio.

Django viene equipado con un ORM (Object-Relational Mapping) robusto y fácil de usar que permite la interacción con la base de datos utilizando modelos de objetos Python, simplificando así la manipulación de datos y eliminando la necesidad de escribir consultas SQL directamente.

Django ofrece una interfaz de administración automática generada dinámicamente, proporcionando a los desarrolladores un panel de administración listo para usar para administrar modelos de base de datos sin necesidad de escribir código adicional.

Para garantizar la seguridad de las aplicaciones, Django incorpora medidas de protección contra ataques de inyección SQL, Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF), ayudando a construir aplicaciones más seguras desde el diseño.

El sistema de manejo de URL de Django es potente y permite definir rutas y asignar vistas a dichas rutas, facilitando la creación de APIs RESTful y la gestión de endpoints.

Django es altamente extensible a través de aplicaciones y paquetes de terceros, y su comunidad activa ofrece una gran cantidad de paquetes que pueden ser integrados para añadir funcionalidades adicionales, siendo los más conocidos Django REST Framework que permite a Django expandir aún más sus capacidades como servicio Backend, o django-allauth que permite incorporar autenticación de usuario con JWT o OAuth 2.0 de manera rápida y sencilla.

Compatible con una gama de bases de datos, incluyendo PostgreSQL, MySQL y SQLite, Django permite elegir el motor de base de datos que mejor se adapte a sus necesidades. Aunque es compatible con motores de bases de datos NoSQL, la compatibilidad directa con los motores de bases de datos relacionales es notable, siendo Django conocido por su fuerte integración con bases de datos relacionales, especialmente a través de su ORM. La utilización de bases de datos no relacionales puede requerir esfuerzos y configuraciones adicionales.

Django es adaptable a proyectos de todos los tamaños, aunque algunas características pueden parecer excesivas para proyectos más simples.

Django aplica el principio de convenciones sobre la configuración (CoC), lo que significa seguir un conjunto de reglas predefinidas que pueden facilitar la configuración, aunque algunos desarrolladores pueden preferir un enfoque más configurable.

En proyectos pequeños o simples, donde la riqueza de características de Django podría ser considerada excesiva, puede ser más eficiente utilizar un framework más ligero y específico. Además, en casos donde se utilicen bases de datos no relacionales, puede ser útil buscar soluciones más centradas en esa tecnología.

Como framework completo, Django puede requerir más recursos que frameworks más ligeros, por lo que si se desarrolla una aplicación con restricciones de recursos significativas, se deben considerar alternativas más livianas.

Flask

Flask es un microframework web para Python que se destaca por su simplicidad y facilidad de uso. Ofrece las funcionalidades básicas necesarias para crear aplicaciones web y APIs, dando a los desarrolladores la libertad de añadir componentes adicionales según sus necesidades.

Flask sigue un enfoque minimalista, proporcionando solo las funcionalidades esenciales para el desarrollo web sin imponer una estructura rígida.

Una característica clave de Flask es la definición de rutas para manejar diferentes métodos HTTP a través de decoradores en funciones, lo que facilita la asociación directa entre rutas y código.

Por defecto, Flask utiliza Jinja2, un motor de plantillas que permite la generación dinámica de contenido HTML, facilitando la separación de la lógica de presentación del código Python. Aunque es posible cambiar el motor de plantillas si así lo requiere el desarrollador.

Flask implementa un sistema de middleware que permite ejecutar funciones durante el ciclo de vida de una solicitud, facilitando la implementación de lógica común como autenticación y manejo de errores.

Flask no exige un sistema de base de datos específico, pero es compatible con varios, incluyendo sistemas SQL y NoSQL. Los desarrolladores tienen la libertad de seleccionar y configurar el sistema de base de datos que mejor se ajuste a sus necesidades, así como la posibilidad de utilizar un ORM si se prefiere.

Flask promueve la limitación de las funcionalidades básicas en favor de un marco de trabajo ligero, permitiendo que el desarrollador añada lo que necesite de acuerdo a sus criterios. Fomenta la incorporación de funcionalidades adicionales a través de extensiones y paquetes, facilitando la personalización de la aplicación de acuerdo a los requisitos específicos del proyecto.

Flask cuenta con una comunidad activa y un ecosistema de extensiones bien establecido. Los desarrolladores pueden beneficiarse de una variedad de herramientas y complementos desarrollados por la comunidad para expandir la funcionalidad de sus aplicaciones.

Flask es una opción excepcional para aquellos que buscan desarrollar aplicaciones web de manera rápida y sin la sobrecarga de un marco más robusto, presentando un enfoque basado en la simplicidad y flexibilidad.

Sin embargo, su uso puede no ser la mejor opción para aplicaciones muy grandes y complejas que requieren un conjunto más amplio de herramientas integradas. Debido a su sencillez y ligereza, puede ser necesario incorporar funcionalidades que estarían incluidas nativamente en otras soluciones más completas.

ExpressJs

Express.js, a veces referido simplemente como Express, es un framework para la creación de servidores web en Node.js, enfocado en el desarrollo de aplicaciones web y la construcción de APIs. Este framework ofrece herramientas para gestionar rutas, middleware, vistas, y otros elementos relacionados con el servidor, siendo muy popular en la comunidad de Node.js por su simplicidad y flexibilidad.

Uno de los rasgos distintivos de Express es su enfoque minimalista y no invasivo, proporcionando un conjunto de funciones básicas y permitiendo a los desarrolladores añadir módulos adicionales de acuerdo a las necesidades específicas de sus aplicaciones.

Express facilita la definición de rutas para gestionar diferentes tipos de solicitudes HTTP, como GET y POST, ofreciendo un enrutamiento sencillo y flexible que facilita la creación de APIs RESTful y la administración de endpoints.

El framework incluye un sistema de middleware que permite a los desarrolladores ejecutar funciones durante el ciclo de vida de una solicitud HTTP, facilitando la implementación de funciones como autenticación, registro de solicitudes y manejo de errores.

Una de las fortalezas de Express.js es su extensa variedad de motores de plantillas disponibles, lo que permite a los desarrolladores elegir la solución que mejor se adapte a sus necesidades y facilita la configuración de la tecnología.

Express proporciona mecanismos sólidos para manejar errores, facilitando la identificación y la gestión de problemas durante el desarrollo y la ejecución de la aplicación.

A diferencia de otros frameworks, Express no impone un sistema de base de datos específico, aunque sí es compatible con varios sistemas de base de datos, permitiendo la integración sencilla con bases de datos SQL y NoSQL. Esto implica que la responsabilidad de configurar la base de datos y el ORM recae en el desarrollador si se desea utilizar uno.

Express cuenta con una comunidad activa y un ecosistema de módulos y middleware extenso, lo que facilita la incorporación de nuevas funcionalidades a través de paquetes de terceros.

En cuanto a desventajas, si se necesita un framework con muchas características integradas, como un ORM incorporado o un sistema de autenticación avanzado, podría ser más adecuado considerar otros frameworks que ofrecen estas características de forma nativa. Sin embargo, Express puede superarse con las configuraciones adecuadas.

Spring Boot

Java con Spring Boot es una combinación potente y ampliamente adoptada para el desarrollo de aplicaciones backend. Basado en Java, que es reconocido por su robustez y versatilidad, Spring Boot cuenta con un ecosistema maduro y establecido, con una gran cantidad de librerías y paquetes, facilitando la integración con otras tecnologías y el desarrollo en general.

Spring Boot, que forma parte del ecosistema Spring, es un framework que simplifica y acelera el desarrollo de aplicaciones Java. Proporciona un conjunto completo de características, incluyendo manejo de dependencias, configuración automática, y desarrollo basado en convenciones. Utiliza intensivamente la inyección de dependencias, lo que facilita la gestión de componentes y la creación de aplicaciones modulares.

La configuración en Spring Boot se realiza principalmente mediante anotaciones y archivos de configuración, simplificando la tarea de personalizar y configurar la aplicación.

Spring Boot ofrece soporte para una variedad de bases de datos, tanto SQL como NoSQL, facilitando la integración con sistemas de almacenamiento de datos. No incluye un ORM por defecto, pero si se desea, debe ser configurado.

Proporciona funcionalidades integradas para implementar medidas de seguridad, como autenticación y autorización.

Aplica el concepto de convenciones sobre configuración (CoC), que consiste en establecer automáticamente configuraciones predeterminadas basadas en convenciones y patrones, en lugar de requerir configuraciones explícitas para cada detalle de una aplicación.

Spring Boot es ideal para proyectos de cualquier tamaño, pero se destaca particularmente en proyectos más complejos donde se beneficia de su conjunto completo de características.

Debido a su amplia utilización, Spring Boot dispone de una amplia gama de herramientas y bibliotecas para facilitar integraciones con otras herramientas y tecnologías, además de para poder obtener ciertas funcionalidades.

Java, de forma nativa, al realizar algún cambio debería volver a compilarse la aplicación para probar el mismo, hay herramientas en Spring boot que solventan esta situación, como Spring Boot DevTools que permiten el "hot reloading" o recarga en caliente.

Java con Spring Boot es una elección sólida para el desarrollo backend, especialmente en proyectos de tamaño medio a grande, donde la modularidad, la seguridad y la facilidad de configuración son aspectos cruciales.

Como punto débil de esta combinación Java - Spring boot es que Java es un lenguaje altamente tipado, y puede ser considerado más verboso en términos de líneas de código necesarias para realizar ciertas tareas, aunque Spring Boot ayuda a simplificar muchas cosas, Java todavía puede requerir más código que algunos lenguajes más concisos.

Base de Datos

La base de datos es fundamental en cualquier proyecto de desarrollo de software ya que sirve como el núcleo central de almacenamiento y gestión de datos. Una base de datos eficiente y bien diseñada permite a las aplicaciones manejar múltiples usuarios simultáneamente, almacenar información de manera precisa y confiable, evitar la redundancia de datos y procesar los datos de formas poderosas y atractivas . Además, las bases de datos pueden escalar a medida que crece un negocio, lo que significa que pueden adaptarse a las demandas cambiantes de los usuarios y del sistema. La elección del sistema de gestión de bases de datos (DBMS) también juega un papel importante, ya que las decisiones de diseño pueden estar influenciadas por la elección del DBMS, permitiendo una implementación más específica y adaptada a las necesidades del proyecto. En resumen, la base de datos es un componente crítico en el desarrollo de software que facilita el manejo de datos y mejora la funcionalidad y escalabilidad de las aplicaciones.

MySQL

MySQL es una base de datos relacional de código abierto reconocida por su confiabilidad y simplicidad, utilizada en numerosas aplicaciones web y empresariales. Se basa en el modelo relacional, organizando los datos en tablas interrelacionadas mediante SQL para consultas y manipulaciones.

Distribuido bajo la GNU General Public License, MySQL es gratuito y permite a los usuarios modificarlo y distribuirlo. Es multiplataforma, funcionando en Windows, Linux, macOS, entre otros sistemas operativos, favoreciendo su versatilidad en diferentes entornos.

Su escalabilidad es notable, permitiendo manejar grandes volúmenes de datos y transacciones, haciéndolo adecuado para entornos de producción. Es ideal para almacenar datos transaccionales y opera con ACID, Atomicidad, Consistencia, Aislamiento, Durabilidad, lo que lo hace preferente para aplicaciones que requieran estas características.

MySQL soporta una amplia gama de tipos de datos, como enteros, decimales, cadenas de texto y fechas, brindando flexibilidad a los desarrolladores. Ofrece herramientas y utilidades de administración para facilitar la configuración, monitoreo y mantenimiento de las bases de datos. También se integra con herramientas de terceros para administración visual y análisis de rendimiento.

La fortaleza de MySQL radica en su comunidad activa y en el extenso ecosistema de herramientas, extensiones y soporte que promueven su adopción y desarrollo. Sin embargo, es importante destacar que, aunque es de código abierto, MySQL es propiedad de Oracle Corporation, lo cual plantea preocupaciones sobre la dirección futura del proyecto y su alineación con la estrategia de Oracle.

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional y objeto-relacional de código abierto que destaca por su solidez, extensibilidad y cumplimiento con los estándares ANSI SQL. Su robustez y capacidad para adaptarse a una variedad de aplicaciones, desde proyectos pequeños hasta grandes infraestructuras empresariales, lo convierten en una elección popular.

Basado en el modelo relacional, PostgreSQL permite estructurar datos en tablas relacionadas y, además, proporciona una gran extensibilidad a través de la creación de tipos de datos y funciones personalizadas. La Licencia PostgreSQL, que es una licencia de código abierto, permite el uso, modificación y distribución gratuita del software.

PostgreSQL es altamente extensible, permitiendo a los usuarios definir sus propias funciones, operadores, tipos de datos y agregados, lo que facilita la personalización según las necesidades específicas de cada proyecto. Desde 2001, PostgreSQL ha ofrecido soporte para transacciones ACID, garantizando la integridad y consistencia de los datos en cualquier circunstancia.

Además de los tipos de datos estándar, PostgreSQL incluye tipos de datos avanzados como hstore, JSON y arrays multidimensionales, lo que brinda flexibilidad para representar una amplia gama de información. Una característica distintiva de PostgreSQL es su soporte para la herencia de tablas, lo que permite la creación de tablas que heredan atributos y métodos de otras tablas, simplificando así la gestión y organización de los datos.

MongoDB es un sistema de gestión de bases de datos NoSQL basado en documentos y de código abierto, desarrollado por MongoDB Inc. Almacena datos en un formato BSON, una representación binaria de JSON, que permite una mayor flexibilidad en la representación de información en comparación con las bases de datos relacionales tradicionales.

Una característica distintiva de MongoDB es que no requiere un esquema fijo para los documentos, lo que significa que cada documento dentro de una colección puede tener un conjunto variable de campos. Esta falta de esquema implica que los documentos pueden adaptarse fácilmente a cambios en la estructura de los datos, proporcionando una gran flexibilidad.

MongoDB

MongoDB es capaz de crear índices para optimizar el rendimiento de las consultas y ofrece un lenguaje de consulta potente que permite operaciones de filtrado, proyección, ordenamiento y agregación en los datos almacenados. Por defecto,

MongoDB ofrece un modelo de consistencia eventual, pero los usuarios pueden configurar niveles de consistencia más estrictos si lo requieren.

Este sistema de base de datos es particularmente adecuado para manejar grandes volúmenes de datos no estructurados y semiestructurados, lo que lo hace popular en escenarios de Big Data. MongoDB tiene una amplia comunidad de usuarios y desarrolladores, y MongoDB Inc. ofrece servicios de soporte empresarial.

MongoDB es reconocido por su escalabilidad horizontal, soporte para datos no estructurados y gestión de grandes volúmenes de información. Sin embargo, cabe mencionar que no proporciona transacciones ACID completas en todos los casos, especialmente cuando se trata de múltiples documentos o colecciones, lo que puede limitar su utilidad en aplicaciones que requieren transacciones fuertemente consistentes.

Despliegue / Hosting

Vercel

Vercel destaca como una plataforma de desarrollo que concentra su atención en el despliegue y la administración del lado del cliente o "frontend" en las aplicaciones web. Su concepción se basa en simplificar la construcción, implementación y escalado de aplicaciones web modernas, especialmente aquellas creadas con JavaScript y frameworks como Next.js.

El enfoque de Vercel en el frontend se justifica por una serie de características distintivas de su plataforma. En primer lugar, proporciona una infraestructura ágil y escalable, capaz de manejar un volumen considerable de solicitudes sin comprometer el rendimiento de la aplicación. Además, ofrece almacenamiento serverless para el frontend, lo que permite a los desarrolladores cargar archivos estáticos directamente en su red de entrega de contenido global, optimizando así los tiempos de carga para los usuarios finales.

Vercel incorpora funciones de borde (Edge Functions), permitiendo la ejecución de código en el servidor sin necesidad de gestionar infraestructuras propias. Esta

capacidad posiciona las funciones de borde en proximidad a los usuarios, lo que potencialmente mejora la velocidad y reduce costos.

Asimismo, la plataforma integra herramientas de desarrollo compatibles con marcos de trabajo populares de frontend, lo que facilita una colaboración eficiente entre equipos de desarrollo. En línea con esto, Vercel prioriza la experiencia del desarrollador (DX), ofreciendo funcionalidades como previsualizaciones de despliegues, integración con sistemas de gestión de contenido y opciones de personalización para adaptarse a las necesidades específicas de cada proyecto.

En cuanto a seguridad y gestión de código, Vercel implementa medidas para proteger las aplicaciones y garantizar un flujo de trabajo efectivo. Su evolución como plataforma de nube orientada al frontend se refleja en la expansión de sus capacidades, como la introducción de bases de datos gestionadas y almacenamiento, demostrando un compromiso continuo con la automatización y la simplificación del proceso de desarrollo.

Digital Ocean

DigitalOcean, destaca como proveedor de servicios de computación en la nube que ofrece una plataforma de Infraestructura como Servicio (IaaS) especialmente diseñada para desarrolladores de software. Reconocida por su popularidad entre la comunidad de código abierto, DigitalOcean compite directamente con gigantes como Amazon Web Services (AWS).

La plataforma de DigitalOcean incluye una variedad de servicios principales. Por ejemplo, los "Droplets" son instancias privadas de máquinas virtuales (VM) que los desarrolladores pueden configurar según sus necesidades, asemejándose a las instancias de Amazon EC2 o Azure, pero exclusivamente compatibles con sistemas operativos Linux. Además, DigitalOcean ofrece una solución de Kubernetes administrado, permitiendo a los usuarios desplegar clústeres de Kubernetes con facilidad.

El "App Platform" es otra destacada oferta de DigitalOcean, proporcionando una plataforma como Servicio (PaaS) que simplifica la publicación de código en los servidores del proveedor y la implementación de aplicaciones en la nube. Por otro lado, la empresa también ofrece servicios de almacenamiento, incluyendo Block

Storage para asignar volúmenes adicionales a los Droplets, y Object Storage para almacenar grandes volúmenes de datos.

DigitalOcean destaca por su facilidad de uso, su disponibilidad de APIs para automatización e integración, y su documentación exhaustiva. Además, la empresa ofrece garantías de tiempo de actividad del 99.99% para VMs y almacenamiento. Sin embargo, es importante tener en cuenta algunas limitaciones, como su enfoque exclusivo en sistemas operativos Linux, opciones de pago que cobran incluso cuando las instancias están apagadas, y una disponibilidad geográfica más limitada en comparación con otros proveedores de la nube.

Render

Render es una plataforma de computación en la nube que se destaca por su enfoque en la simplicidad y la eficiencia en el despliegue y la administración de aplicaciones modernas. Render ofrece una variedad de servicios para desarrolladores y equipos de desarrollo, permitiéndoles construir, implementar y escalar aplicaciones en la nube de manera rápida y sencilla.

Una de las características principales de Render es su facilidad de uso. La plataforma se centra en la eliminación de la complejidad asociada con la configuración y administración de infraestructuras, permitiendo a los desarrolladores enfocarse en la creación de aplicaciones de alta calidad en lugar de perder tiempo en tareas operativas. Render simplifica el proceso de implementación con un enfoque en la automatización y la integración continua, lo que facilita el despliegue de aplicaciones con solo unos pocos clics o comandos en la línea de comandos.

Además de su enfoque en la simplicidad, Render ofrece una amplia gama de servicios que cubren las necesidades de desarrollo y despliegue de aplicaciones modernas. Estos servicios incluyen alojamiento web estático y dinámico, gestión de bases de datos, servidores de aplicaciones, funciones sin servidor y más. Render se integra con tecnologías populares como Docker, Node.js, Python, Ruby on Rails, y muchos otros, lo que permite a los desarrolladores trabajar con las herramientas y lenguajes que mejor se adapten a sus necesidades.

Otro aspecto destacado de Render es su enfoque en la transparencia y la previsibilidad en la facturación. La plataforma ofrece una estructura de precios clara y

transparente, sin costos ocultos ni tarifas sorpresa. Los usuarios solo pagan por los recursos que consumen, lo que les permite controlar y prever sus costos de infraestructura de manera efectiva.

En resumen, Render es una plataforma de computación en la nube que se distingue por su enfoque en la simplicidad, la automatización y la transparencia en la facturación. Con su amplia gama de servicios y su enfoque en la facilidad de uso, Render es una opción atractiva para desarrolladores y equipos de desarrollo que buscan una solución simple y eficiente para construir, implementar y escalar aplicaciones en la nube.

API

REST

Representational State Transfer (REST) es un estilo arquitectónico introducido por Roy Fielding en el año 2000 como parte de su tesis doctoral, concebido para sistemas hipermedios distribuidos. Aunque no es un protocolo ni un estándar formal, REST ha ganado prominencia como una metodología para el diseño de APIs web.

El enfoque de REST se fundamenta en un conjunto de restricciones y principios que buscan simplificar el diseño y la escalabilidad de sistemas distribuidos, haciendo énfasis en la ausencia de estado. Una interfaz de servicio se considera RESTful cuando cumple con seis principios fundamentales.

El principio de Cliente-Servidor establece una separación clara entre la interfaz de usuario y la interfaz de programación de aplicaciones, permitiendo su evolución independiente. Por otro lado, la restricción de Stateless dicta que cada solicitud del cliente debe contener toda la información necesaria para que el servidor pueda entenderla y procesarla, sin mantener ningún estado del cliente entre peticiones.

La capacidad de cacheo, otro principio de REST, permite que las respuestas del servidor indiquen si pueden ser almacenadas en caché, lo que puede aumentar la eficiencia al disminuir la carga en el servidor. La uniformidad de la interfaz, a su vez, se logra mediante URIs bien definidos, métodos HTTP estándar y mensajes autoexplicativos, simplificando la interacción con la API.

REST también promueve la organización en capas de los componentes de la API, facilitando la separación de responsabilidades y la modularidad del sistema. Finalmente, aunque no es obligatorio, el principio de Código en Demanda permite que el servidor proporcione código ejecutable al cliente para extender su funcionalidad, como applets de Java o scripts de JavaScript.

Es esencial destacar que, si bien HTTP es comúnmente utilizado para implementar REST, este estilo arquitectónico no está limitado exclusivamente a HTTP y puede ser aplicado a otros protocolos. La clave reside en los principios y no en la implementación específica.

Las APIs RESTful adheridas a estos principios facilitan la interoperabilidad entre diferentes aplicaciones y sistemas, lo que permite a los desarrolladores construir servicios web escalables y mantenibles que promueven la eficiencia y la flexibilidad en la comunicación entre sistemas distribuidos.

GraphQL

GraphQL es un lenguaje de consulta y una especificación desarrollada por Facebook en 2012 y lanzada como código abierto en 2015. A diferencia de REST, que se basa en operaciones CRUD (Crear, Leer, Actualizar, Borrar) predefinidas, GraphQL ofrece a los clientes la capacidad de solicitar sólo los datos que necesitan, lo que lo hace más flexible y eficiente en entornos donde se requieren consultas específicas y optimizadas.

La principal característica de GraphQL es su capacidad para permitir a los clientes especificar exactamente qué datos necesitan, lo que elimina el exceso de datos transmitidos sobre la red y mejora el rendimiento de las aplicaciones. Esto se logra mediante la definición de un único punto de entrada (endpoint) para la API y el uso de un sistema de tipado fuerte que describe la estructura de los datos disponibles.

Otra ventaja de GraphQL es su capacidad para unificar múltiples fuentes de datos en una sola consulta, lo que simplifica el desarrollo de aplicaciones que requieren datos de diferentes fuentes. Además, GraphQL proporciona herramientas para la validación de consultas y la introspección de esquemas, lo que facilita la comprensión y el mantenimiento de las API GraphQL.

Aunque GraphQL es una tecnología poderosa y versátil, no es adecuada para todos los casos de uso. En situaciones donde la estructura de los datos es simple y predecible, o donde se requiere una alta disponibilidad en caché de los datos, REST puede ser una mejor opción. Sin embargo, para aplicaciones complejas con requisitos de consulta flexibles y variados, GraphQL ofrece una solución más adecuada y eficiente.

gRPC

gRPC es un sistema de comunicación de código abierto desarrollado por Google en 2015. Se basa en el protocolo HTTP/2 para la transferencia eficiente de datos y utiliza el formato de serialización de mensajes Protocol Buffers (protobuf) para la definición de interfaces y la comunicación entre clientes y servidores.

Una de las características distintivas de gRPC es su soporte para la comunicación bidireccional de streaming, lo que permite la transmisión simultánea de datos desde el cliente al servidor y viceversa. Esto hace que gRPC sea especialmente adecuado para aplicaciones en tiempo real, como chat en línea, juegos multijugador y transmisión de medios.

gRPC utiliza un sistema de generación de código para crear stubs de cliente y servidores a partir de archivos de definición de interfaz escritos en Protocol Buffers. Esto simplifica el desarrollo de aplicaciones al proporcionar una interfaz de programación fuertemente tipada y auto documentada.

Además, gRPC ofrece características avanzadas como autenticación, autorización y cifrado de extremo a extremo, lo que garantiza la seguridad y la integridad de las comunicaciones entre clientes y servidores.

Aunque gRPC es una tecnología poderosa y eficiente, no es adecuada para todos los casos de uso. Por ejemplo, en situaciones donde la interoperabilidad con sistemas legacy es crucial, REST puede ser una mejor opción debido a su amplia adopción y soporte en una variedad de plataformas y lenguajes de programación.

Generación de Reportes y Dashboards

Metabase

Metabase es una herramienta de análisis de datos de código abierto diseñada para simplificar y democratizar el acceso a los datos dentro de las organizaciones. Fundada en 2015 y basada en San Francisco, Metabase ofrece una solución intuitiva y fácil de usar que permite a los usuarios explorar y visualizar datos de manera efectiva sin necesidad de conocimientos técnicos avanzados.

Una de las principales características de Metabase es su capacidad para conectarse a una amplia variedad de fuentes de datos, incluyendo bases de datos relacionales como MySQL, PostgreSQL y SQL Server, así como bases de datos NoSQL como MongoDB y Google BigQuery. Además, Metabase puede integrarse con servicios en la nube populares como Amazon Redshift, Google Analytics y Salesforce, lo que permite a los usuarios acceder y analizar datos de múltiples fuentes en un solo lugar.

Una vez conectado a los datos, Metabase ofrece una amplia gama de herramientas de análisis y visualización que permiten a los usuarios explorar y comprender sus datos de manera efectiva. Esto incluye la capacidad de crear consultas SQL personalizadas, generar gráficos y tablas interactivas, y compartir visualizaciones con otros miembros del equipo a través de informes y paneles de control.

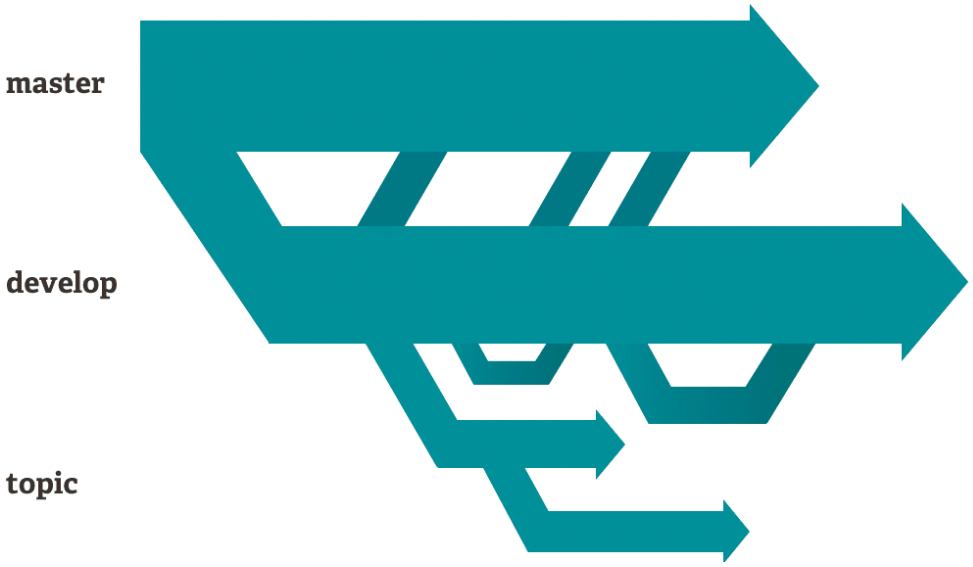
Además de su funcionalidad de análisis y visualización, Metabase también ofrece características avanzadas de administración y colaboración que facilitan la gestión y el intercambio de datos dentro de las organizaciones. Esto incluye la capacidad de establecer permisos granulares para controlar quién puede acceder a qué datos, así como la capacidad de programar y automatizar informes para garantizar que los usuarios siempre tengan acceso a la información más reciente.

En resumen, Metabase es una herramienta poderosa y fácil de usar que está diseñada para democratizar el acceso a los datos dentro de las organizaciones. Con su amplia compatibilidad con diferentes fuentes de datos y su conjunto de características intuitivas de análisis y visualización, Metabase es una opción atractiva para cualquier organización que busque mejorar su capacidad para explorar y comprender sus datos.

Herramientas para control de versión de software

GitHub

GitHub es una plataforma de desarrollo de software basada en Git que proporciona alojamiento de repositorios remotos, así como una variedad de otras herramientas de colaboración para programadores. Los repositorios en GitHub son contenedores para proyectos de software, y cada repositorio contiene archivos y directorios que se pueden editar, descargar, subir y compartir.



GitHub resulta una herramienta indispensable en el desarrollo ya que permite la colaboración entre varios desarrolladores. Utilizando GitHub, es posible que varios desarrolladores trabajen en el mismo proyecto de manera simultánea, a la vez que cada uno puede tener su propia rama de desarrollo.

Otra gran herramienta proporcionada por GitHub es la capacidad de implementar integración continua, donde se pueden automatizar pruebas, construcción y despliegue de aplicaciones.

GitLab

GitLab es una plataforma de desarrollo de software de código abierto que ofrece control de versiones y colaboración similar a GitHub. Al igual que GitHub, GitLab se basa en Git y proporciona funcionalidades para la gestión de repositorios, la integración continua, la gestión de problemas y la colaboración entre equipos.

Propuesta de Solución

Una vez propuesta y evaluada nuestra problemática principal, y habiendo repasado las soluciones similares, las tecnologías y frameworks disponibles, llegamos a la selección del siguiente conjunto de tecnologías, teniendo en cuenta además las capacidades individuales y la posibilidad de trabajar en grupo.

- Flutter, SDK para el desarrollo multiplataforma de la aplicación con la que interactúan tutores y alumnos.
- React, una biblioteca para interfaces de usuario web, brinda la interfaz para que los coordinadores interactúen con el programa.
- Django, un marco de trabajo para Python que proporciona una sólida base para el backend.
- Metabase, para el análisis de datos y la evaluación del estado del proyecto.
- Render, plataforma en la nube que permite desplegar y administrar aplicaciones web y servicios de backend.
- Digital Ocean, otra plataforma en la nube similar a Render
- Vercel, una plataforma de desarrollo que se centra principalmente en el despliegue y la gestión del lado del cliente o "frontend" de las aplicaciones web.

La elección de Flutter como marco de desarrollo para la aplicación móvil se debe a que cuenta con uno de los mayores soportes para la construcción de aplicaciones

multiplataforma. En nuestro caso, tenemos que desarrollar la aplicación de forma tal que la puedan usar la mayor cantidad de estudiantes y alumnos en sus dispositivos móviles, por lo tanto, no podemos desarrollar solo para Android o solo para iOS. Es verdad que podríamos tomar una aproximación de desarrollo nativa en ambos entornos y ecosistemas, pero eso implica tener que aprender dos lenguajes distintos y conocer a fondo los dos entornos.

Flutter nos permite reducir el estudio de un marco de desarrollo móvil a uno solo, ya que puede compilar para ambos dispositivos desde el mismo código fuente. Además, la estructura de código en Flutter es una de las más amigables con nuestro conocimiento, ya que es orientada a objetos y muy similar a C++. Por otra parte, revisando la biblioteca de paquetes disponibles en el repositorio de pub.dev, encontramos muchos módulos y paquetes ya construidos que solucionan varios de los aspectos que serán necesarios implementar en la aplicación, como puede ser una vista de calendario en la aplicación móvil para tutores y alumnos, o botones que permiten calificar una tutoría.

Elegimos React para el desarrollo del tablero web debido a que ya habíamos trabajado un poco con esa librería, lo que nos brindaba un concepto general sobre la estructura del código fuente y las posibles opciones a considerar a la hora de buscar paquetes de UI. Además, conseguimos que este tablero esté alojado en la web a través de la utilización de Next.js, un framework de React que nos permite crear aplicaciones web de alta calidad con renderizado en el lado del servidor y generación de sitios estáticos, lo que facilita el despliegue y la escalabilidad de nuestra aplicación. En conjunto con bibliotecas de componentes como *shadcn*, con muchas menos horas de desarrollo podemos tener un gran tablero web, que brinde una buena experiencia de usuario, y enfocarnos en el manejo de datos y solicitudes a la base de datos.

La elección de Django para desarrollar el servicio de Backend se debe principalmente a que contamos con experiencia en Python y Django posee un enfoque centrado en incrementar la eficiencia del desarrollador y promover la creación de código limpio y reutilizable, así como la posibilidad de integrar módulos de terceros que realizan partes del flujo de datos que no necesitamos desarrollar desde cero. Estas características en conjunto nos permitieron desarrollar de manera rápida y eficaz el servicio de backend sin dejar de lado aspectos importantes como la seguridad y calidad del código.

Para la Base de Datos optamos por utilizar PostgreSQL. Los motivos para esta decisión son principalmente dos: En primer lugar, el programa TAIE tiene una naturaleza relacional, es

decir, todo dentro del programa mantiene conexiones y congruencia entre los diferentes componentes del mismo. En segundo lugar, PostgreSQL ofrece un sólido soporte para operaciones relacionales y una amplia gama de funcionalidades que satisfacen las necesidades de nuestro proyecto. Además, su robustez y escalabilidad nos brindan confianza en cuanto a la gestión de los datos a largo plazo.

Para el despliegue del backend de nuestra aplicación, utilizamos Render ya que ofrece una integración fluida con Django, lo que facilita el proceso de despliegue y mantenimiento. Render también ofrece una extensa documentación sobre cómo realizar el despliegue de una aplicación de Django y vincularla con la base de datos PostgreSQL, lo que nos permite realizar un despliegue simple y funcional en poco tiempo. Además, la plataforma Render proporciona herramientas de monitoreo y escalabilidad que nos permiten gestionar eficientemente el rendimiento de nuestra aplicación en producción.

Finalmente, utilizamos Metabase para otorgar a los miembros de la UCC encargados de coordinar y monitorear el programa TAIE una herramienta clara y simple pero potente para generar dashboards

Alcance funcional

En este apartado se definen las principales funciones de nuestro sistema, que incluyen la gestión de horarios, la programación de tutorías, la inscripción de tutores, y la evaluación y seguimiento del progreso del programa.

A partir de los conocimientos previos sobre cómo funciona el programa, y nuestra entrevista con Maria Orozco, coordinadora principal del programa TAIE, que nos brindó los principales requerimientos para los modelos roles presentes actualmente en el programa, generamos las siguientes historias de usuario.

Definimos las historias de usuario utilizando el siguiente formato:

COMO **SUJETO** QUIERO **ACCIÓN** PARA **OBJETIVO**

y las clasificamos según el sujeto.

Estudiante (tutorando)

- Como estudiante, quiero poder registrarme en la aplicación para solicitar una tutoría con un tutor del programa TAIE.
- Como estudiante, quiero poder iniciar sesión a través de Google, para no tener que recordar contraseñas y utilizar mi cuenta institucional.
- Como estudiante, quiero poder seleccionar mis carreras al momento de registrarme, para ver los horarios disponibles de mis materias.
- Como estudiante, quiero poder ver los horarios disponibles de los tutores en la aplicación para poder reservar o unirme a un turno en un horario conveniente para mí.
- Como estudiante, quiero poder buscar los horarios disponibles en función de los temas y conceptos que definen a las distintas áreas de conocimiento disponibles, para reservar un horario con un tutor que dicte esas clases.

- Como estudiante, quiero ver una lista de mis tutorías próximas, para volver a consultar el estado y detalles de las mismas.
- Como estudiante, quiero poder darme de baja en una tutoría próxima en la que haya decidido anotarme, para informar que no asistiré.
- Como estudiante, quiero ver los detalles de la tutoría a la que me inscribí, para poder saber su estado y actuar acorde al mismo.
- Como estudiante, quiero poder postularme como tutor, para participar del programa TAIE en calidad de tutor en caso de ser aprobado.
- Como estudiante, quiero poder elegir la fecha en la cual reservar el horario de la tutoría, para recibir apoyo del tutor de acuerdo a mis necesidades de tiempo.
- Como estudiante, quiero poder elegir el área que necesito aprender, para que la clase reservada se dicte según mis necesidades.
- Como estudiante, quiero poder inscribirme a un horario ya reservado, para unirme a una tutoría que explique un tema que me interesa en un horario conveniente.
- Como estudiante, quiero poder ver un calendario que ordene mis reservas de tutorías, para poder organizarme y entender mi cronograma.
- Como estudiante, quiero que las instancias se muestren con un color distintivo según su estado, para saber el estado de la misma sin necesidad de abrir el detalle.
- Como estudiante, quiero poder acceder a la reunión virtual desde mi dispositivo en caso de que la tutoría sea virtual, para no tener que coordinar el enlace con el tutor.
- Como estudiante, quiero poder enviar una revisión sobre la instancia de tutoría en la cual me anoté, para brindar comentarios y evaluar la clase.
- Como estudiante, quiero ver una lista de reportes de tutorías pendientes, para poder seleccionar sobre qué instancia quiero enviar mi revisión.

Tutor

- Como tutor del programa TAIE, quiero poder ingresar en la aplicación para ofrecer mis servicios como tutor.
- Como tutor, quiero poder iniciar sesión a través de Google, para no tener que recordar contraseñas y utilizar mi cuenta institucional.
- Como tutor, quiero ver de forma ordenada y detallada mis horarios, para controlar mi disponibilidad en el programa.
- Como tutor, quiero poder agregar un nuevo horario disponible, indicando el día de la semana, la modalidad, los horarios de inicio y fin, y la capacidad de alumnos que

tendrá este horario, para ofrecer mi disponibilidad y permitir que los alumnos reserven turnos según sus necesidades.

- Como tutor, quiero poder modificar un horario ya creado, para no tener que crear uno nuevo en caso de que sea necesaria una modificación.
- Como tutor, quiero poder eliminar un horario, para que no se muestre disponible para reserva en caso de que no me sea conveniente ofrecer ese horario.
- Como tutor, quiero poder ver un calendario que ordene mis instancias de tutorías, para poder organizarme y entender mi cronograma.
- Como tutor, quiero que las instancias se muestren con un color distintivo según su estado, para saber el estado de la misma sin necesidad de abrir el detalle.
- Como tutor, quiero ver los detalles de las instancias de tutoría en mi calendario, para poder conocer el estado actual y la cantidad de alumnos que asistirán.
- Como tutor, quiero poder cambiar los estados de la instancia de tutoría, para poder brindar feedback a todos aquellos inscritos sobre lo que está sucediendo realmente con la clase.
- Como tutor, quiero poder cancelar una clase, para informar a todos los alumnos que la clase no se dictará.
- Como tutor, quiero poder demorar una clase, para informar a todos los alumnos que mi asistencia no será en punto.
- Como tutor, quiero poder comenzar y concluir una clase, para informar a los alumnos el estado y transcurso de la misma a medida que ocurre.
- Como tutor, quiero poder enviar un reporte sobre una clase que haya dictado previamente y se encuentre concluida, para brindar comentarios y los temas abordados en la clase.
- Como tutor, quiero ver una lista de reportes de tutorías pendientes, para poder seleccionar sobre qué instancia quiero enviar mi reporte.
- Como tutor, quiero poder acceder a la vista de estudiante, para poder utilizar la aplicación con todas las funcionalidades de un estudiante y reservar turnos en caso que lo necesite.

Coordinador

- Como coordinador del programa TAIE, quiero poder ingresar en el tablero web para poder llevar un seguimiento del programa.
- Como coordinador, quiero poder iniciar sesión a través de Google, para no tener que recordar contraseñas y utilizar mi cuenta institucional.
- Como coordinador, quiero poder ver el listado de postulaciones de los alumnos que quieren ser tutores, para poder conocer más sobre los alumnos postulantes.
- Como coordinador, quiero poder ver los detalles de la postulación, para conocer los datos del alumno postulante y en qué áreas se está postulando.
- Como coordinador, quiero poder seleccionar en qué áreas de conocimiento autorizar al alumno postulante, para mantener un control sobre la distribución de tutores por área de conocimiento.
- Como coordinador, quiero poder adjuntar un comentario al momento de aprobar o rechazar una postulación, para indicar los motivos de mi decisión.
- Como coordinador quiero poder rechazar una postulación, para evitar que alumnos no calificados sean tutores, o que la cantidad de alumnos sea demasiada.
- Como coordinador quiero poder aprobar una postulación, para así poder asignar nuevos alumnos tutores a las diversas áreas.
- Como coordinador, quiero ver un listado de las tutorías que poseen al menos un reporte o una reseña, para realizar un seguimiento sobre las mismas.
- Como coordinador quiero poder acceder a las reseñas de una tutoría, para poder obtener retroalimentación de los alumnos sobre la instancia y el tutor.
- Como coordinador quiero poder acceder a los reportes de una tutoría, para poder obtener retroalimentación del tutor y conocer qué temas son los más solicitados.

Administrador

- Como administrador, quiero poder dar de alta perfiles de secretarios de grado para que puedan acceder con su nuevo rol de coordinador a la aplicación.

Metodología de trabajo

Para poder avanzar en los distintos módulos del sistema, donde las actualizaciones en los modelos del backend afectan la generación de código por parte de los generadores de modelos del frontend móvil, fue necesario idear una estrategia de trabajo colaborativa que permitiera la coordinación entre el frontend y el backend.

Para implementar esta solución, utilizamos GitHub Actions, una herramienta de integración continua que nos permitió automatizar el despliegue del nuevo código.

Previo al paso a paso, es necesario entender que contamos con 3 repositorios principales alojados en GitHub:

- Repositorio del frontend móvil, que contiene la aplicación construida con Flutter y Dart.
- Repositorio del frontend web, que contiene la aplicación construida con React y NextJS.
- Repositorio del backend, que contiene las rutas de la API, así como la comunicación con la base de datos a través del ORM.

Con la finalidad de hacer más fluido el trabajo, la comunicación y el desarrollo de los diferentes módulos, optamos por la implementación de un flujo de CI/CD a través del uso de Github Actions, lo que permitió automatizar la sincronización del estado de las rutas de la API que manejan los modelos.

Paso a paso

El flujo de trabajo se puede describir en los siguientes pasos:

1. Después de desarrollar una nueva funcionalidad para el servicio de backend y probar el funcionamiento de la misma, se actualiza el código en el repositorio remoto de GitHub, en la rama específica para este estado, la cual denominamos “test”.

2. Cada vez que se recibe una actualización en esta rama del repositorio, ocurren 2 cosas en paralelo:
    a. Se realiza un nuevo deploy en Render, para disponibilizar al servicio móvil y al servicio web las nuevas funcionalidades desarrolladas.
    b. Se envía una notificación al repositorio remoto del servicio móvil, indicando que existe una nueva versión del código del servicio API/servidor.

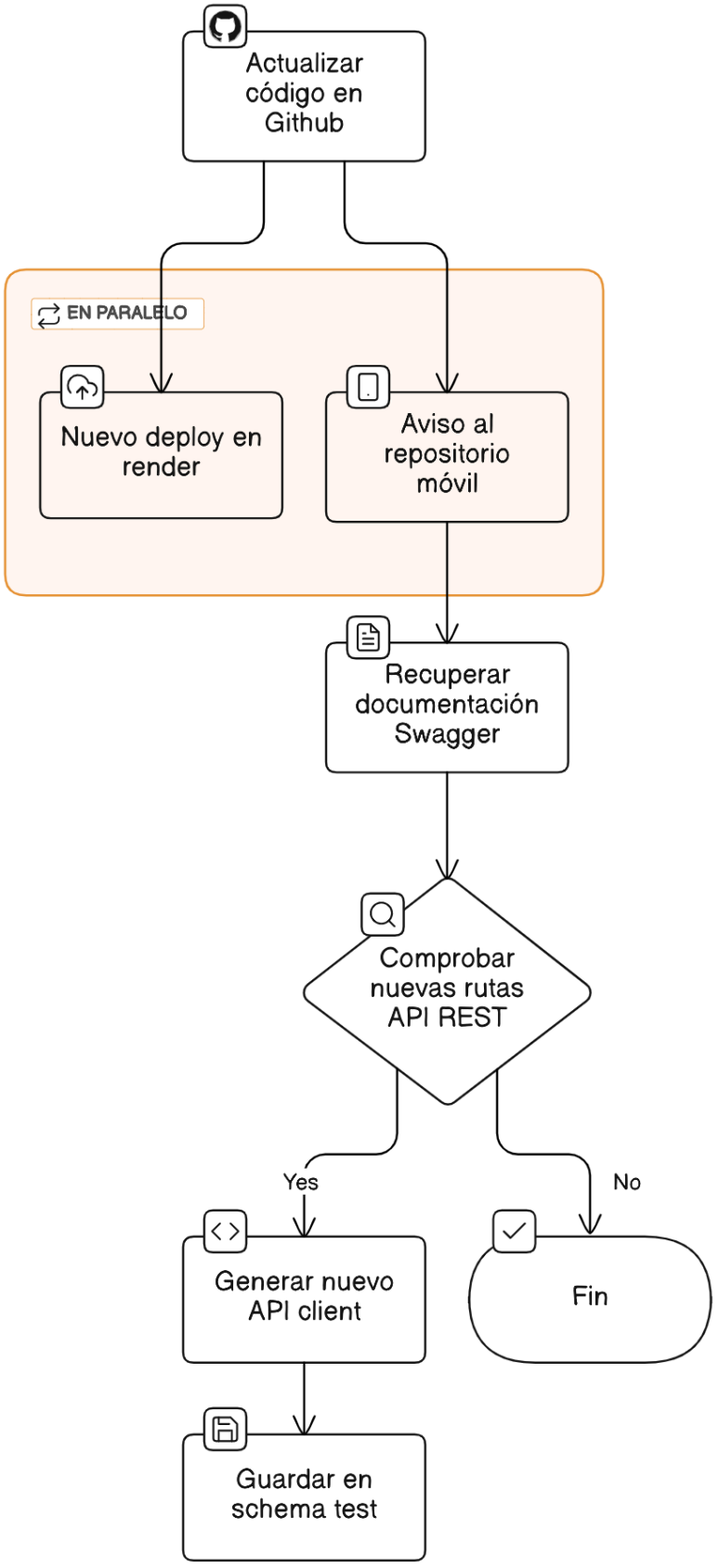
3. El servidor remoto del servicio móvil recupera la documentación generada por Swagger en formato OpenAPI del servidor del servicio API/Servidor y comprueba si las nuevas funcionalidades exponen nuevas rutas API REST que necesiten ser mapeadas dentro de la aplicación móvil.

4. En caso de que existan nuevas rutas, o que las existentes hayan sufrido modificaciones, el repositorio del servicio de API/Servidor, a partir del documento de formato OpenAPI, genera automáticamente un nuevo API Client. Esto nos permite como desarrolladores comunicarnos dentro de la aplicación móvil con nuestra API/Servidor de manera más simple, directa y evitando errores, ya que expone funciones con los datos esperados y sus tipos, así como las posibles respuestas a las mismas.

5. Una vez generado el nuevo API Client, este se guarda en una rama específica del repositorio, la cual denominamos “schema-test”, donde podemos obtener el valor actualizado, sin alterar las demás versiones, que podrían ser estables o no.

Esta metodología de trabajo colaborativo y automatizado nos permite centrarnos en la creación de nuevos módulos y servicios, y la mejora continua del producto.

Actualización y Despliegue de Nuevas Funcionalidades



Solución generada

Para abordar los objetivos generales, específicos y las necesidades particulares de cada rol en el sistema de turnero, proponemos como solución un modelo que integra diversos componentes de nuestra pila tecnológica seleccionada previamente. Este modelo está estructurado de manera cohesiva, donde los diferentes modelos trabajan entre sí para alcanzar un sistema informático tipo turnero, robusto y eficiente.

Usuarios del sistema

Usuario Alumno no registrado

Es un usuario que posee un mail institucional, pero que todavía nunca ha iniciado sesión en la aplicación móvil Turnero TAIE. Sus acciones están limitadas al inicio de sesión y registro dentro de la aplicación. Una vez que un alumno no registrado completa los pasos de registro que recibe en su primer inicio de sesión, cambia su rol a Alumno.

Usuario Alumno

Se trata de un usuario registrado en el sistema, que tiene permisos para hacer uso de la aplicación Turnero TAIE únicamente desde la perspectiva de alumno. Las funciones permitidas desde un usuario alumno son:

- Buscar horarios.
- Agendar y darse de baja en tutorías.
- Generar reseñas de tutorías en las que fueron vinculados.
- Postularse a tutores.

Usuario Tutor

Se trata de un usuario registrado en el sistema al cual, luego de atravesar el proceso de postulación, fue aprobado con éxito por parte de un coordinador y habilitado a brindar tutorías sobre áreas autorizadas. Las funciones permitidas de un usuario tutor son:

- Crear horarios personales para brindar tutorías
- Dar inicio, retrasar o cancelar tutorías
- Generar reportes de sus tutorías finalizadas

Usuario Coordinador

Se trata de un usuario registrado en el sistema, idealmente un miembro UCC del programa TAIE. Las funciones permitidas de un usuario coordinador son:

- Revisar postulaciones de usuarios alumnos.
- Aprobar o rechazar postulaciones de alumnos.
- Autorizar las áreas sobre las que el nuevo usuario tutor puede brindar tutorías.
- Consultar los reportes y reseñas de las tutorías
- Acceder a los reportes y dashboard del programa TAIE

Módulos importados

Los módulos son fragmentos y librerías de código bien definidos que encapsulan funcionalidades específicas y que pueden ser reutilizados en diferentes partes de una aplicación. Nos proveen de soluciones funcionales que podemos readaptar a nuestras necesidades sin la necesidad de incurrir en un gran desarrollo.

Al utilizar paquetes de terceros como estos, aprovechamos las soluciones probadas y confiables desarrolladas por la comunidad.

Módulos de Flutter

Los módulos son importados en el archivo pubspec.yaml, directo de pub.dev.

A continuación, mencionamos aquellos que representan una funcionalidad clave:

- cupertino_icons: Proporciona los iconos de estilo iOS para ser utilizados en el desarrollo de las interfaces de usuario. Esto permite uniformidad en la misma compilando en Android o en iOS.

- sign_in_button: Ofrece un conjunto de botones de inicio de sesión predefinidos que siguen las directrices de diseño de Google y Apple.

- flutter_secure_storage: Esta librería nos permite almacenar datos sensibles (en nuestro caso, los tokens de refresco y de acceso) de manera segura en el dispositivo. Utiliza Keychain en iOS y KeyStore en Android.

- google_sign_in: Proporciona una forma sencilla de integrar la autenticación de Google en aplicaciones Flutter. En conjunto con sign_in_button, nos permite registrar e iniciar sesión de usuario dentro de la aplicación utilizando la autenticación de Google.

- swagger_dart_code_generator: Uno de los componentes más importantes de la aplicación, el generador de código de Swagger crea código Dart (utilizable en nuestra aplicación de Flutter) a partir de documentos Swagger/OpenAPI, permitiendo que el frontend móvil consuma APIs RESTful.

- chopper_generator: Similar a Swagger, genera código Dart para los clientes Chopper a partir de archivos de especificación OpenAPI.

- table_calendar: Proporciona un calendario interactivo que permite la selección de fechas y rangos de fechas. Este es el calendario principal tanto para los alumnos como para los tutores.

- flutter_rating_bar: Es un widget de barra de clasificación que permite a los usuarios dar una puntuación. Es una forma creativa y simple de que los alumnos indiquen la utilidad de la clase.

- url_launcher: Permite lanzar URLs en el navegador web predeterminado del dispositivo o abrir mapas en una aplicación de mapas. En conjunto con las tutorías virtuales, permite a tutores y estudiantes acceder fácilmente a recursos en línea.

Módulos de React

- shadcn: Colección de componentes reutilizables para interfaces de usuario que se pueden implementar directamente en aplicaciones web.
- axios: Biblioteca de JavaScript que se utiliza para hacer solicitudes HTTP desde el navegador o Node.js útil para consumir APIs RESTful.
- next-auth: Es un paquete para agregar autenticación a aplicaciones React que utilizan Next.js, facilita el manejo de la autenticación y sesiones con diferentes proveedores como Google y OAuth.
- react-hook-form: Es una biblioteca para React que implementa la administración de formularios con hooks de React.
- zod: Es una biblioteca de validación de esquemas para JavaScript y TypeScript que se puede usar en aplicaciones React para validar los datos del formulario y garantizar que cumplan con un esquema específico antes de enviarlos.

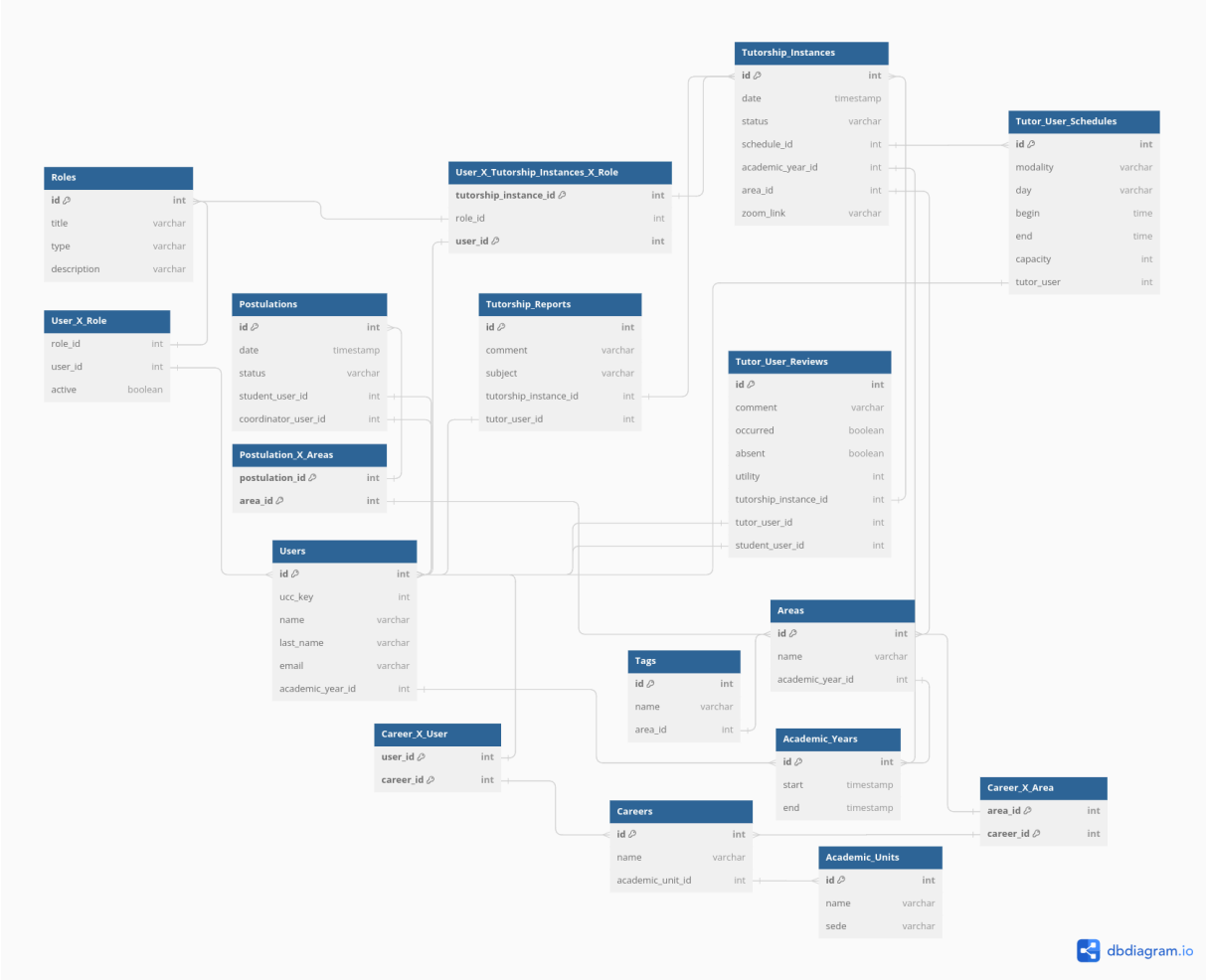
Módulos de Django

Los siguientes módulos mencionados son los paquetes utilizados en nuestro proyecto de Django.

- dj-database-url: Este paquete permite utilizar URL de base de datos en formato URI como variables de entorno, lo cual es útil para configurar fácilmente la conexión a la base de datos en diferentes entornos.
- django-cors-headers: Proporciona un middleware para manejar el CORS. Manejar este recurso de manera efectiva permite que los recursos sean accesibles desde dominios diferentes al del servidor.
- djangorestframework: Toolkit que permite construir APIs web en Django que siguen los principios REST.
- djangorestframework-simplejwt: Proporciona soporte para JWT (JSON Web Tokens) en Django Rest Framework. Es útil para la autenticación y autorización en APIs
- drf-spectacular: Genera documentación OpenAPI para Django Rest Framework, permitiendo que las APIs sean explorables e interactivas.

Diagrama de base de datos

A continuación se detalla el diagrama de la base de datos, la cual es el lugar donde se almacenan los datos de toda la aplicación:



Pantallas del sistema

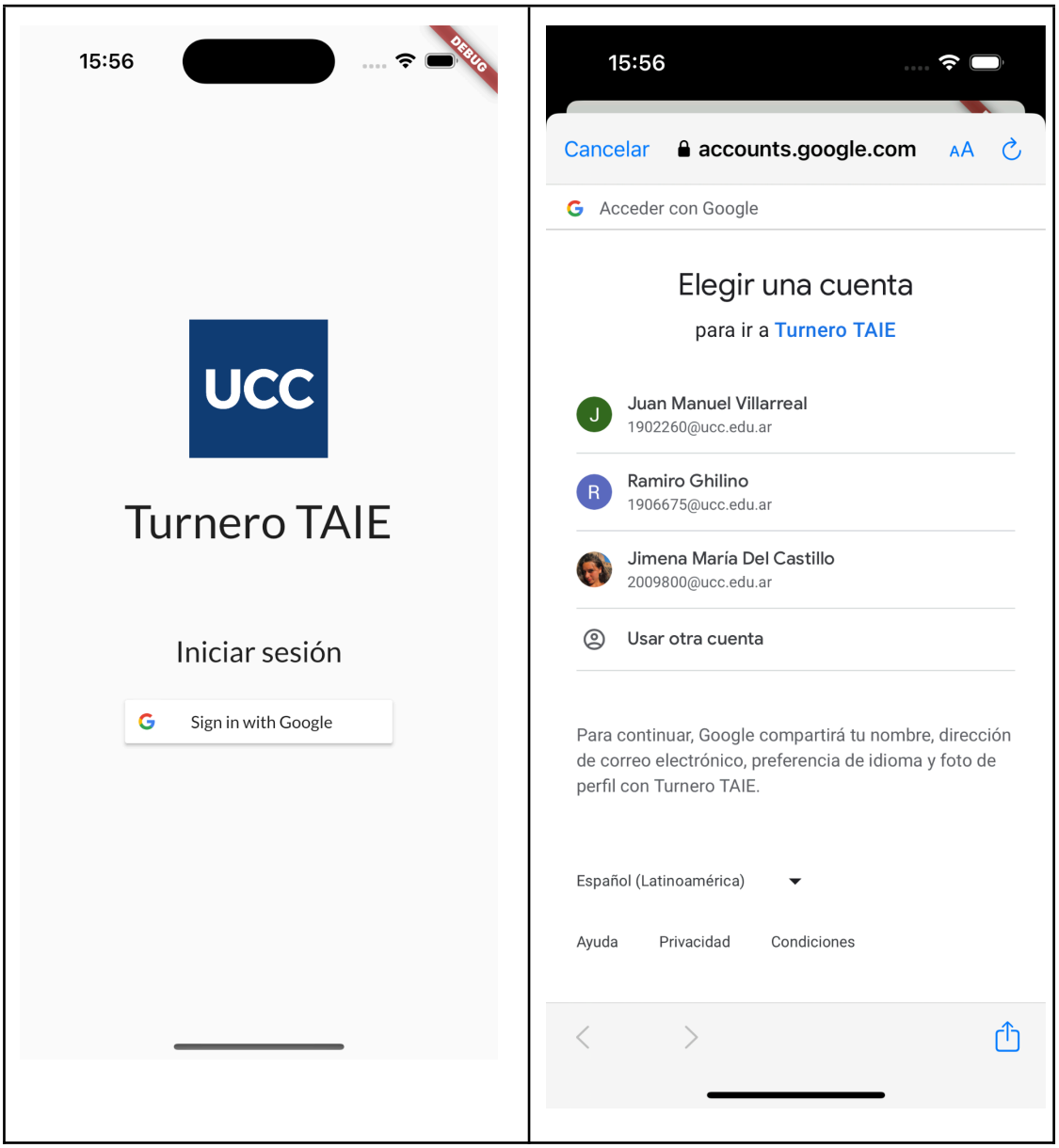
Con el fin de ilustrar las diversas funcionalidades e interfaces de nuestro sistema final, adjuntamos imágenes de aquellas con características centrales que aportan más valor. Cada imagen está acompañada de una breve descripción que explica el flujo de cómo se llega allí y cuáles son las opciones para el usuario.

Aplicación móvil “Turnero TAIE”

Inicio de sesión

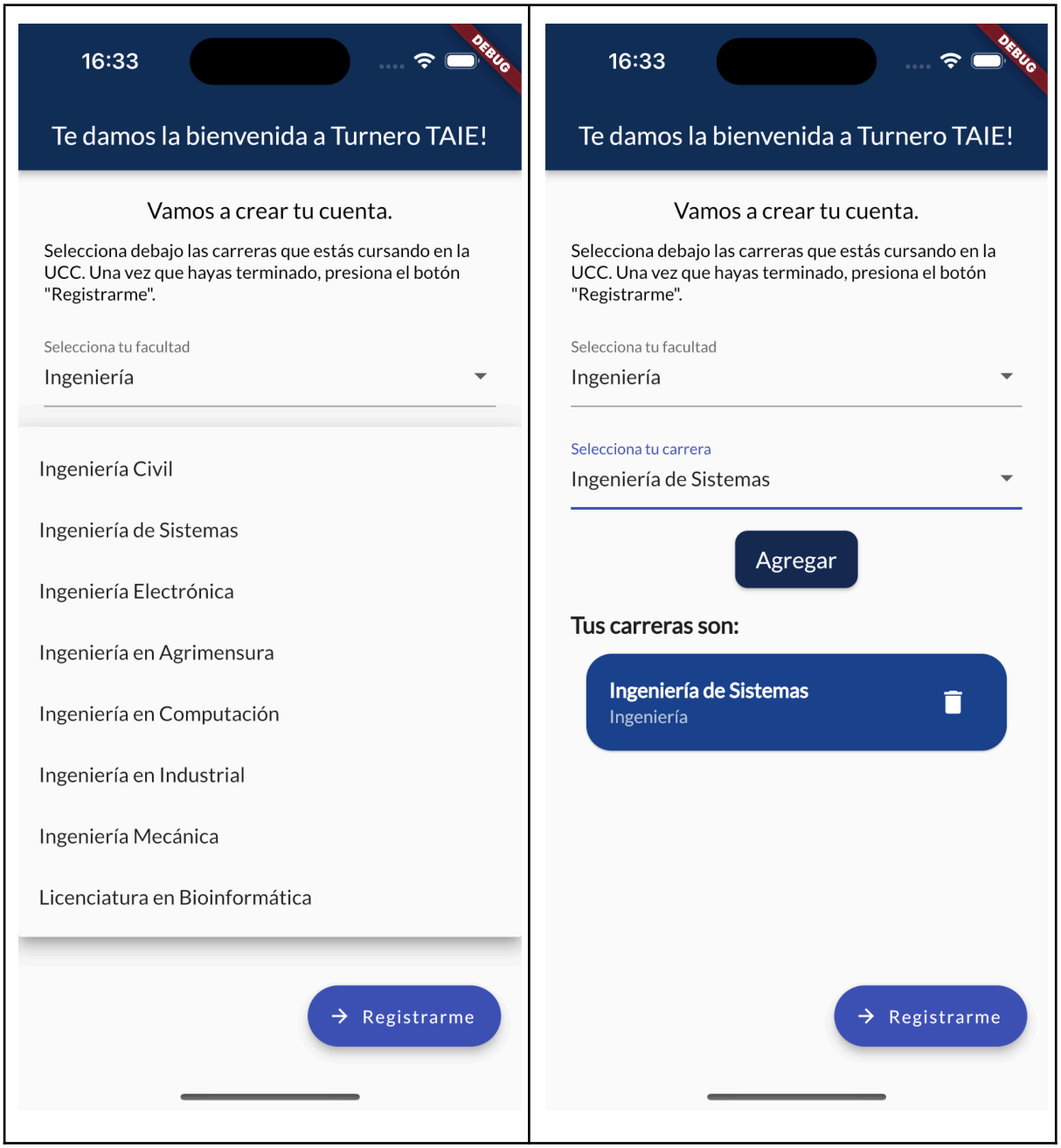
Esta pantalla recibe al usuario al momento de abrir la aplicación, permitiendo iniciar sesión con Google. Allí se le solicitará utilizar una cuenta institucional para iniciar

sesión, ya que mostrará una alerta y volverá a la página principal en el caso de que se desee utilizar otra cuenta.



Registro

La siguiente pantalla permite al usuario alumno seleccionar aquellas carreras a las cuales pertenece, en caso de estar cursando más de una. Solo debe seleccionar la unidad académica correspondiente y posteriormente elegir su carrera. Una vez registrado, llega a la pantalla principal.

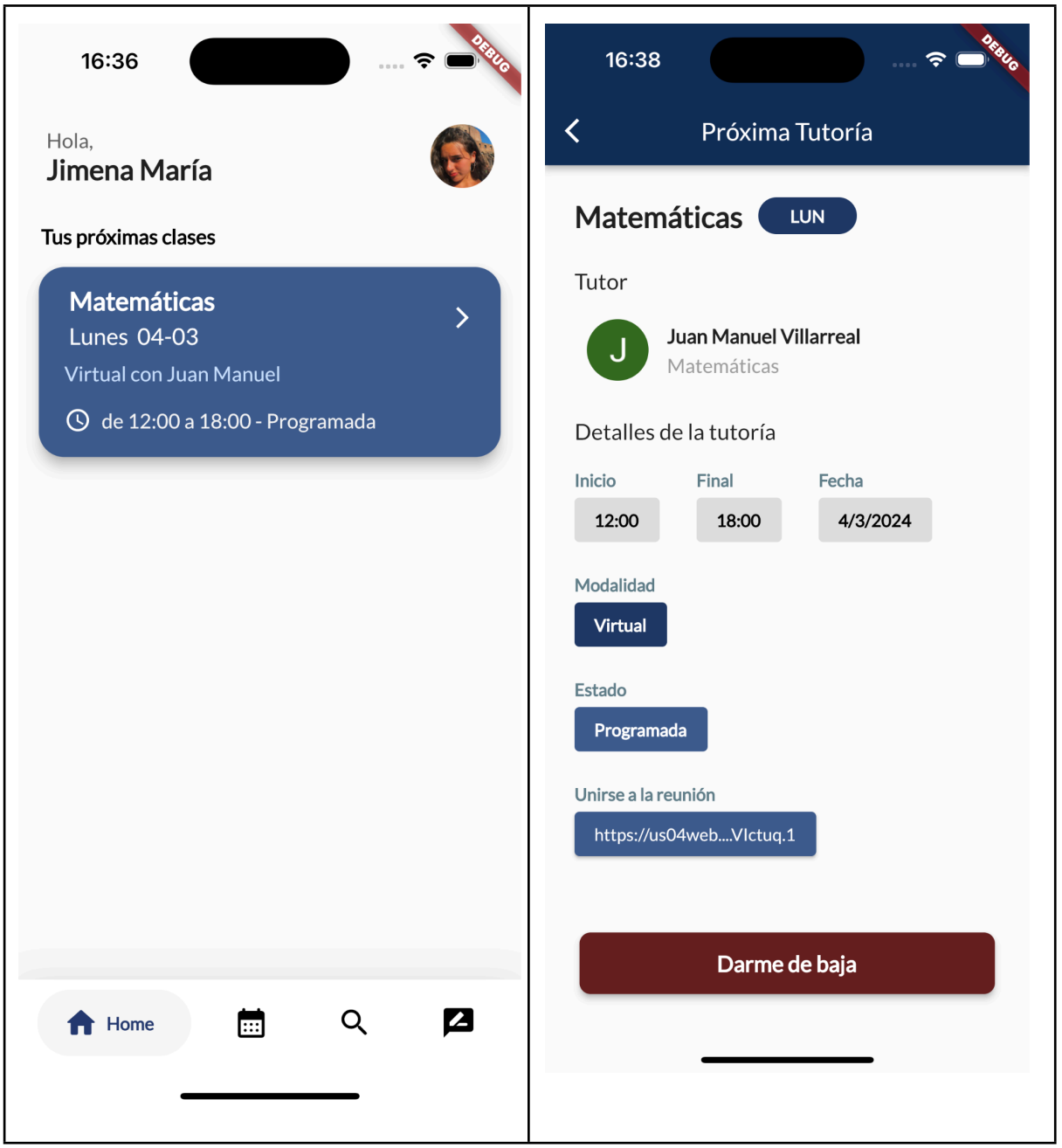


Alumno

Pestaña HOME

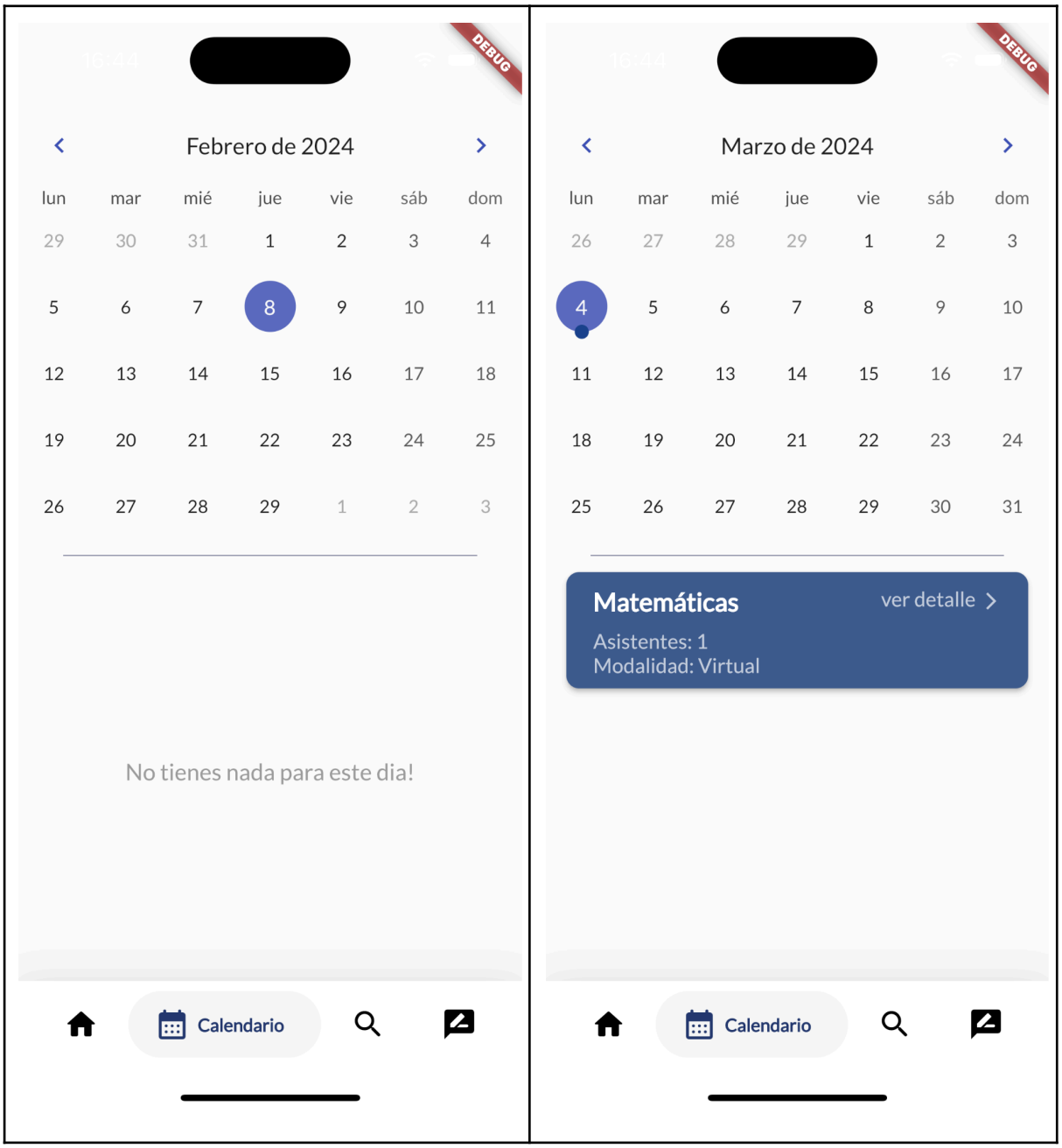
En la pantalla principal encontramos:

- Nombre y foto de perfil del usuario, obtenibles a través del inicio de sesión de Google.
- "Tus próximas clases", donde se listan a través de tarjetas presionables las distintas clases a las que el alumno se inscribe.
- Barra de navegación, que permite desplazarse entre las principales funcionalidades de la aplicación: Home, Calendario, Buscar y Evaluar.



Pestaña CALENDARIO

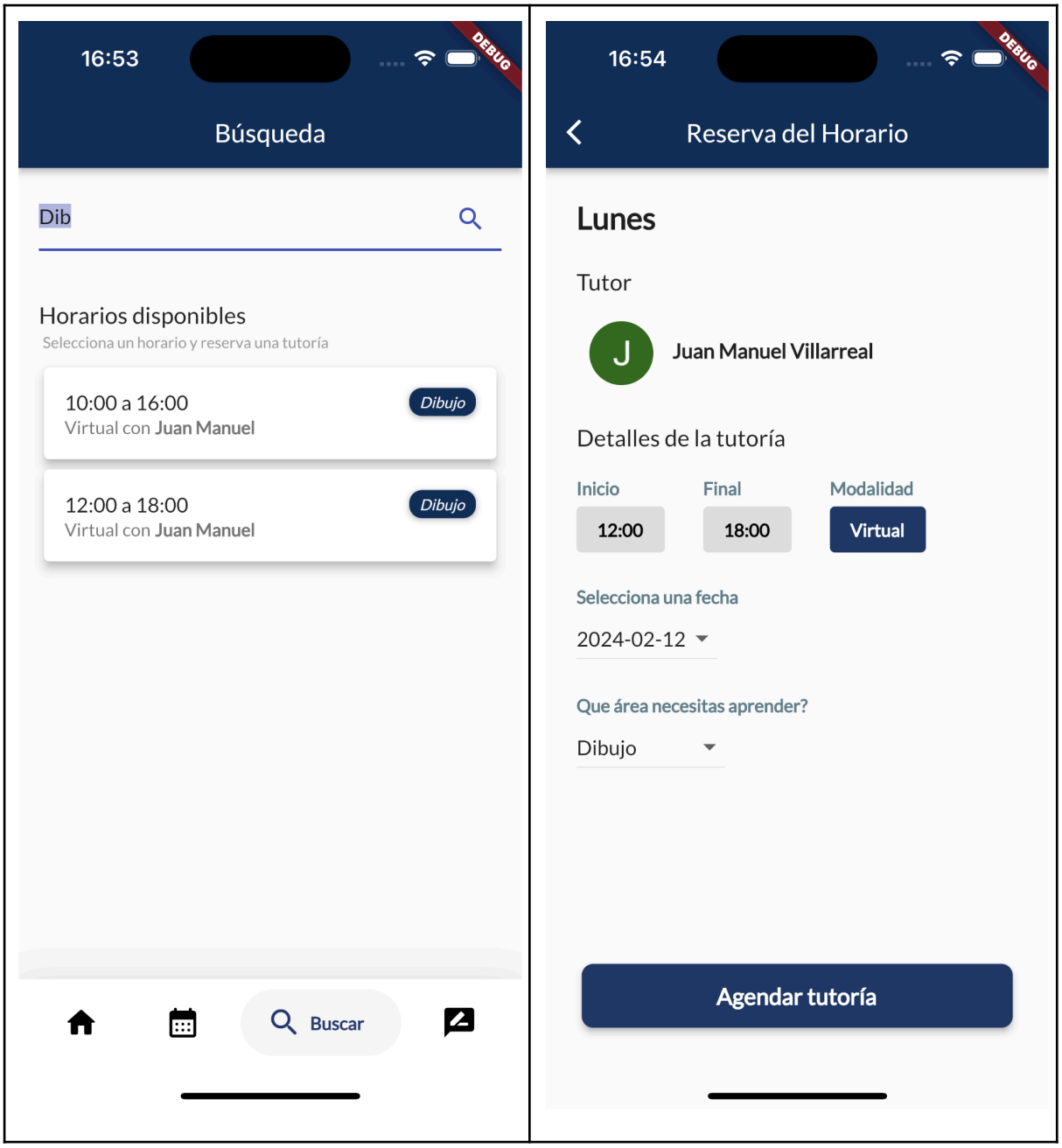
En la página de calendario del alumno, se proporciona rápidamente la información sobre el cronograma de ese usuario alumno. Es posible navegar a través de los meses, ver las distintas instancias señaladas en el calendario y, una vez seleccionado el día, abrir el detalle de esa instancia.

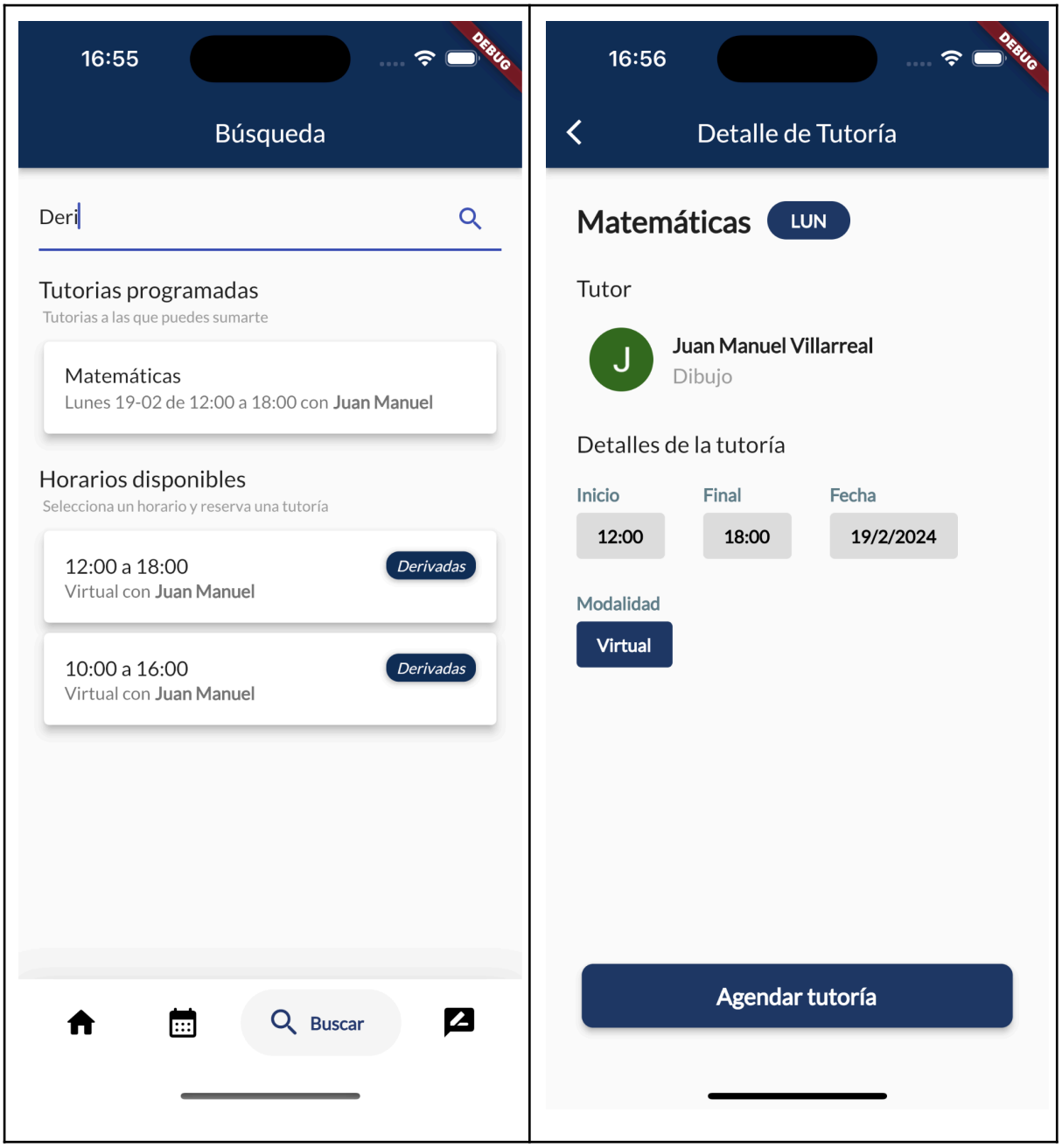


Pestaña BUSCAR

En esta pestaña, el alumno puede buscar materias para inscribirse. La búsqueda se realiza de manera instantánea cada vez que el alumno deja de escribir en el campo de búsqueda. Se pueden buscar tanto por el nombre de la materia como por el área de conocimiento designada o por su etiqueta relacionada.

Si el alumno decide elegir un horario disponible, puede seleccionar el área de conocimiento que desea tratar en la clase, así como también el día dentro de las próximas 5 semanas. En caso de que existan clases ya reservadas, el alumno puede sumarse al grupo e inscribirse a una tutoría ya programada.


16:53
DEBUG
Búsqueda
Dib
Horarios disponibles
Selecciona un horario y reserva una tutoría
10:00 a 16:00
Virtual con Juan Manuel
Dibujo
12:00 a 18:00
Virtual con Juan Manuel
Dibujo
Buscar
16:54
DEBUG
Reserva del Horario
Lunes
Tutor
J
Juan Manuel Villarreal
Detalles de la tutoría
Inicio
Final
Modalidad
12:00
18:00
Virtual
Selecciona una fecha
2024-02-12
Que área necesitas aprender?
Dibujo
Agendar tutoría



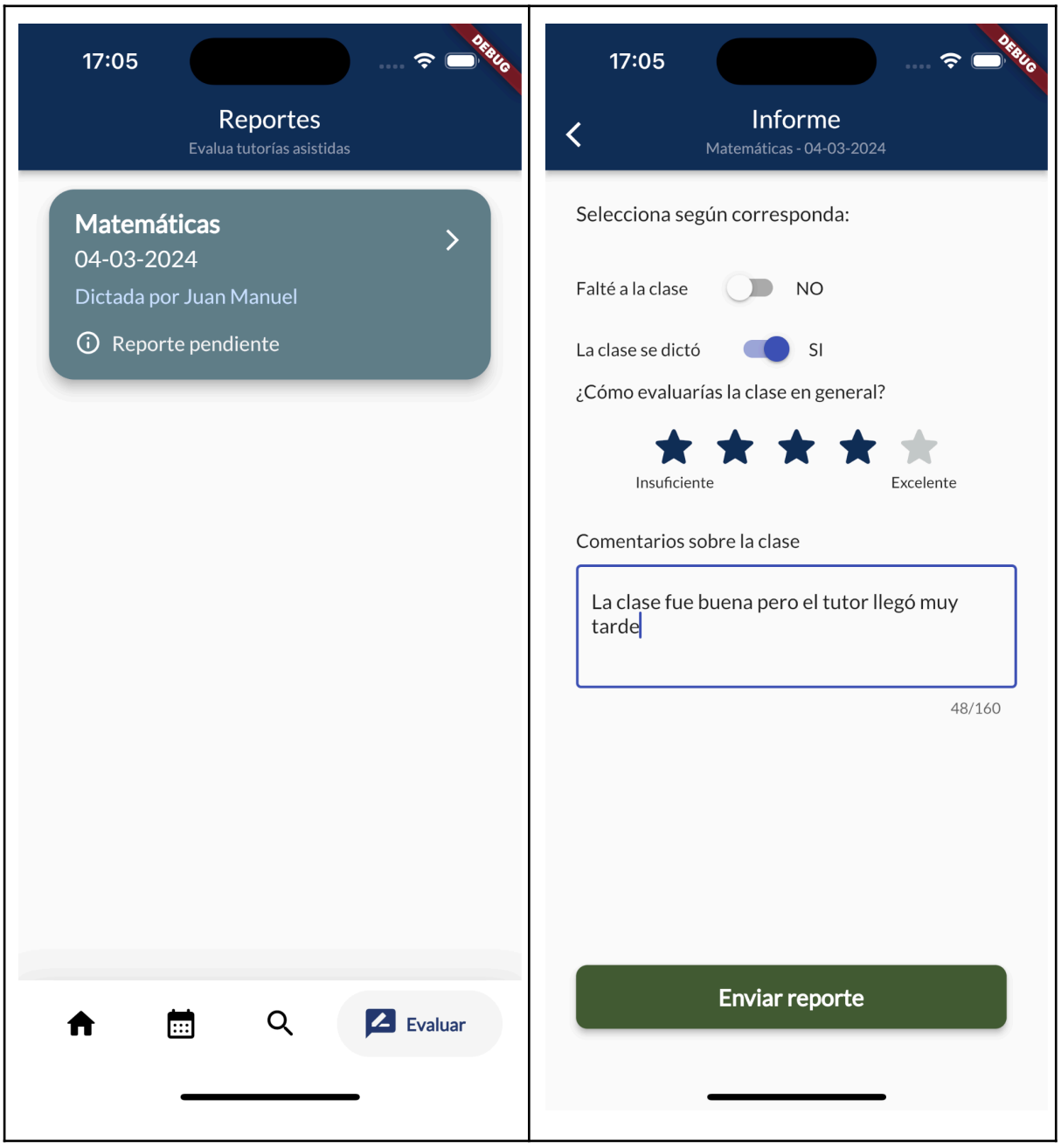
Pestaña EVALUAR

Esta pestaña permite al alumno evaluar las tutorías a las que ha asistido. Se generan tarjetas según la cantidad de revisiones pendientes que tenga ese alumno, con detalles iniciales antes de navegar a la pestaña del informe.

Una vez en el detalle del informe, es posible indicar si el alumno faltó a la clase o si la clase no se dictó a través de dos interruptores.

En caso de que el alumno haya faltado a la clase, el resto de los campos están bloqueados, y puede enviar el informe indicando que no asistió.

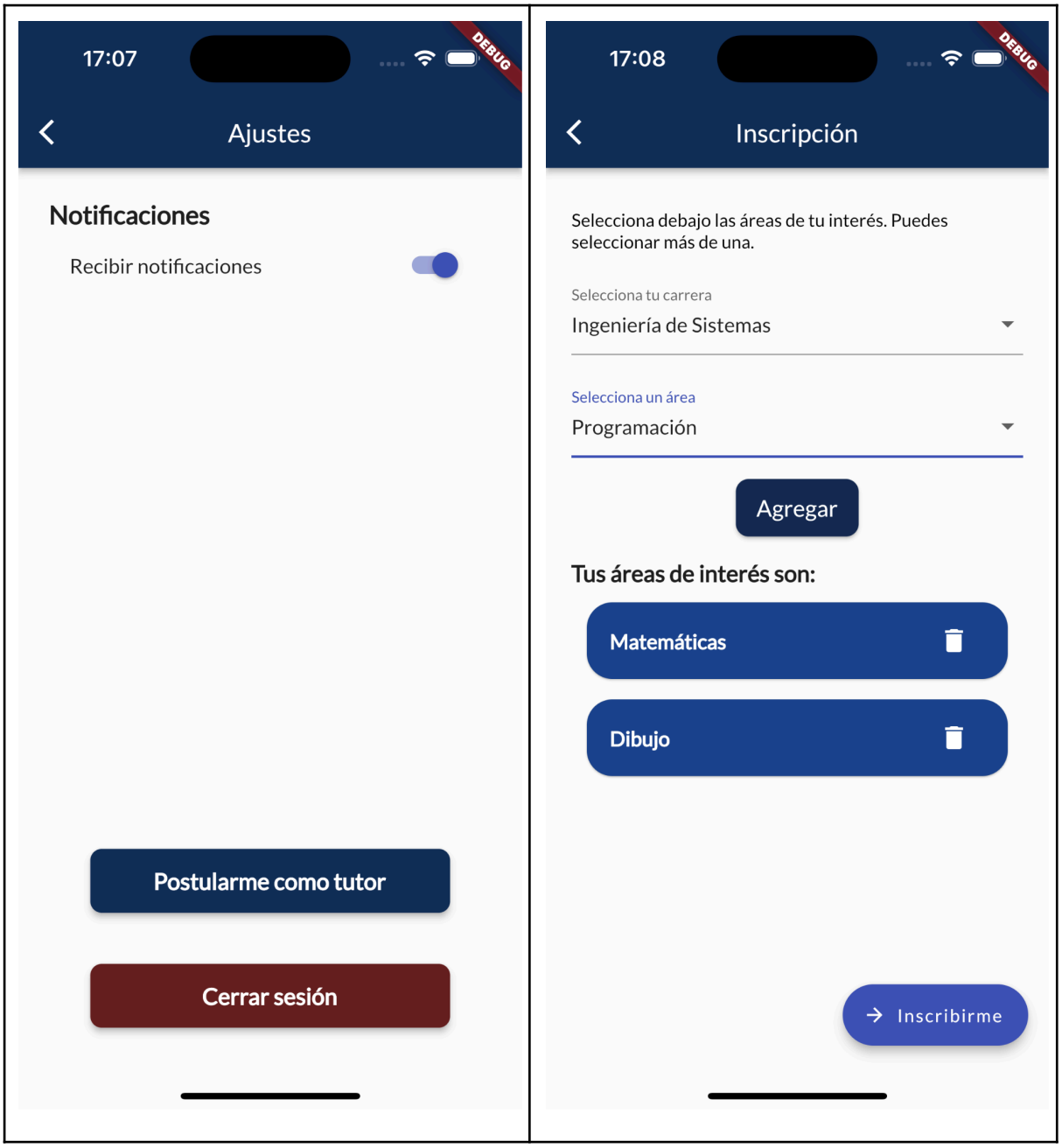
El alumno puede evaluar la clase del 1 al 5 utilizando un sistema de rating con estrellas interactivo, y escribir comentarios sobre la clase que serán vistos por los coordinadores al revisar el informe.



Perfil

Para acceder al perfil, se debe hacer clic en la foto de perfil de la pestaña principal. En esta sección, encontramos dos botones principales: "Cerrar sesión", en caso de que se desee utilizar otra cuenta en este dispositivo, y el botón de "Postulación".

El botón de "Postulación" redirecciona a la página de Inscripción, donde el alumno puede indicar las áreas de interés en las cuales le gustaría participar en calidad de tutor.

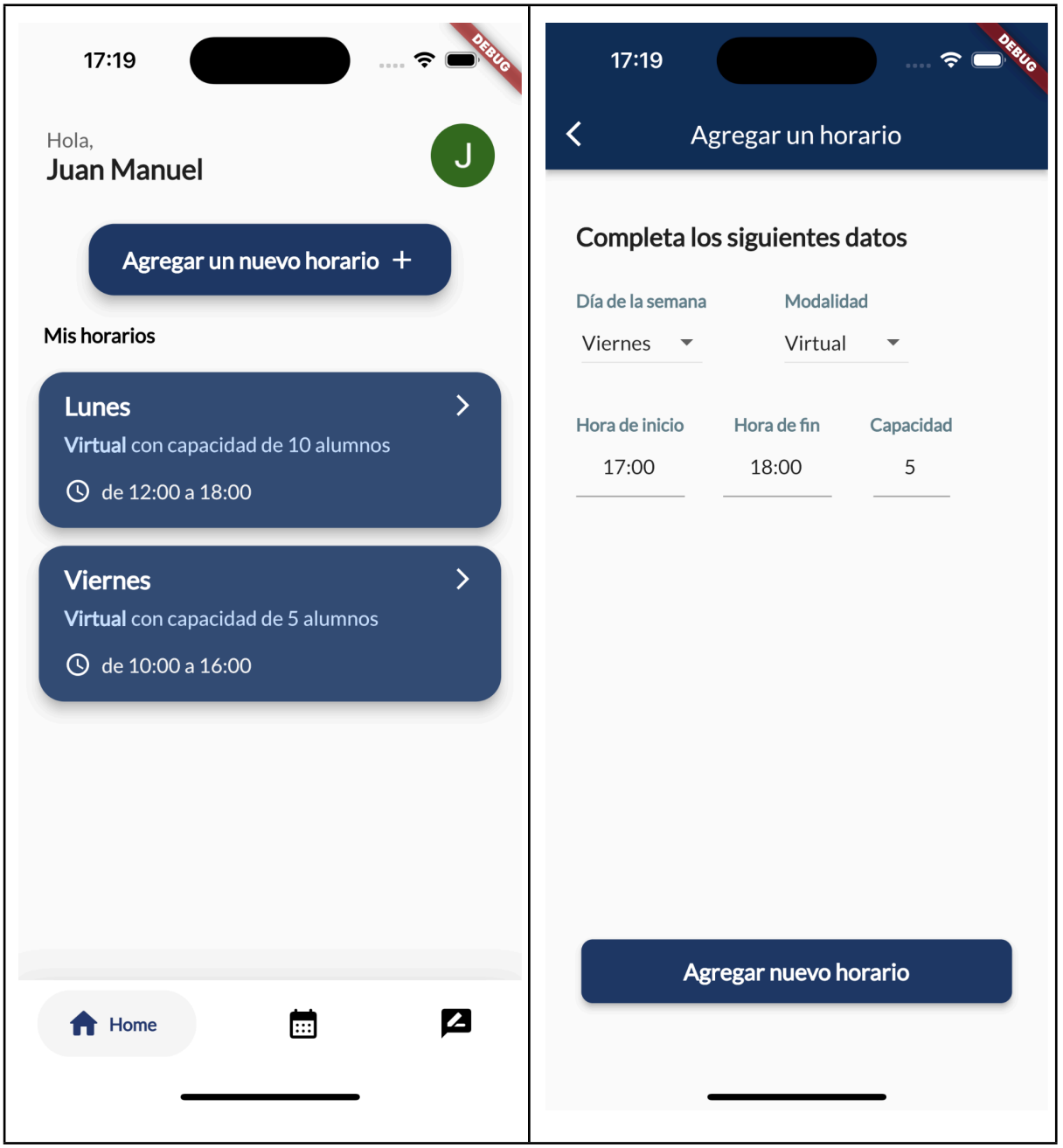


Tutor

Pestaña HOME

Como tutor, la actividad principal es agregar horarios para que los alumnos interesados puedan reservar uno y así tener una tutoría sobre el tema que elijan. Por lo tanto, en la pantalla principal del tutor, se muestra un resumen de los horarios del tutor y un botón para agregar nuevos horarios en caso necesario.

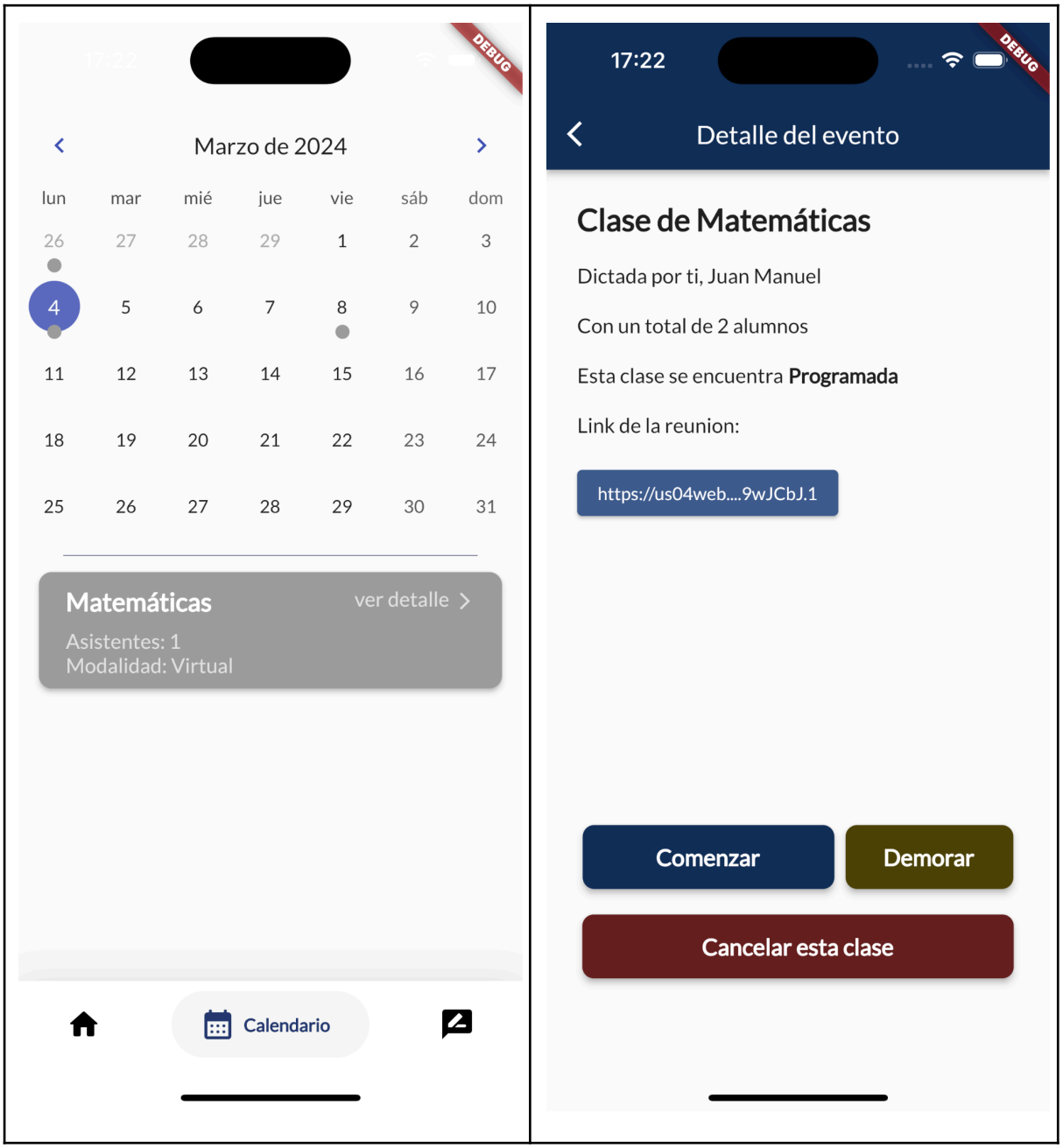
Es posible modificar y eliminar los horarios haciendo clic en las tarjetas correspondientes a los mismos.



Es posible modificar y eliminar los horarios haciendo click en las tarjetas correspondientes al mismo.

Pestaña CALENDARIO

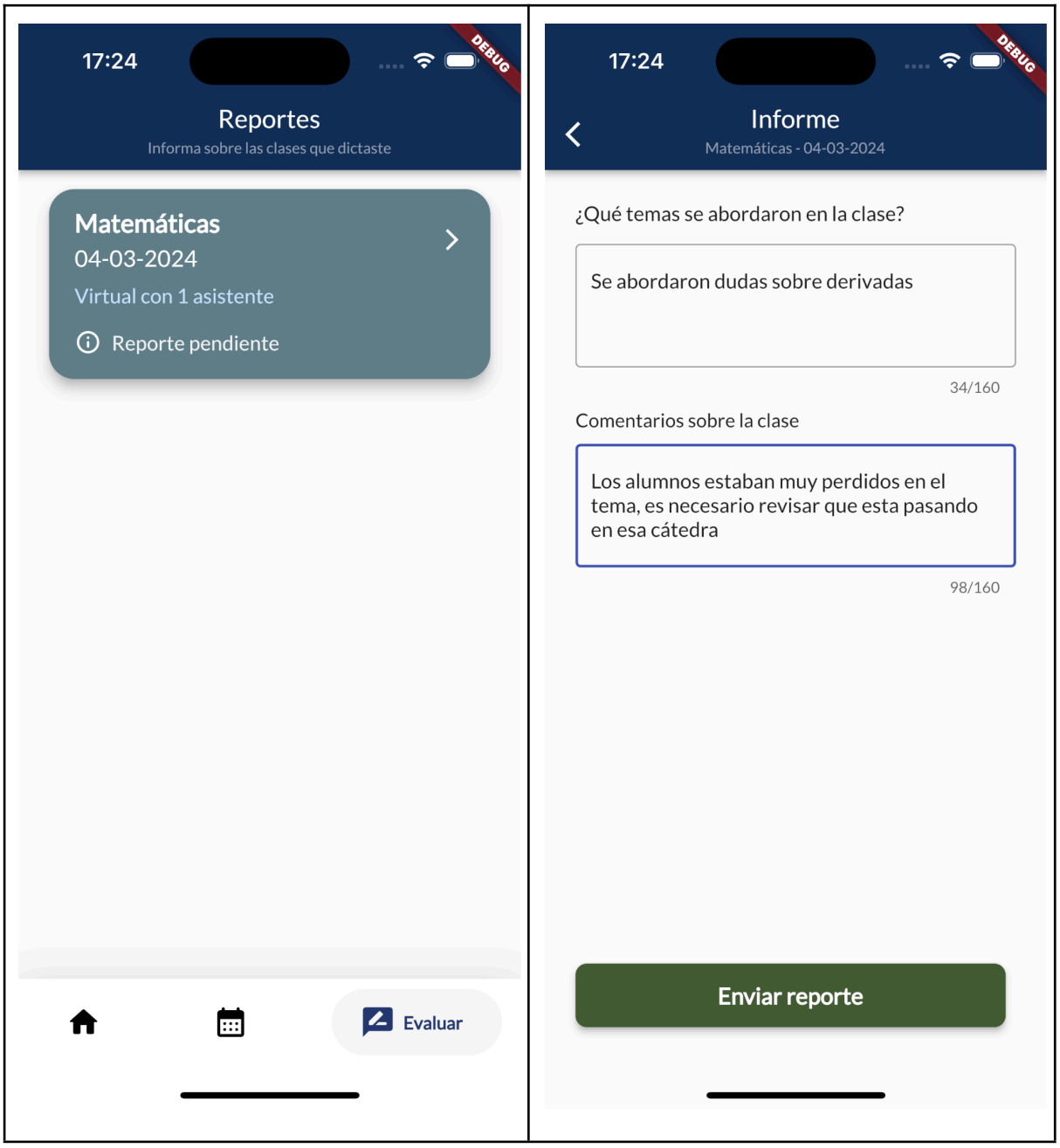
Funciona de la misma manera que para el alumno, permitiendo ver las instancias en sus distintos estados. Desde esta pestaña, el tutor controla los estados de las distintas instancias, lo que le permite avisar si estará demorado o si la clase será cancelada.



Pestaña EVALUAR

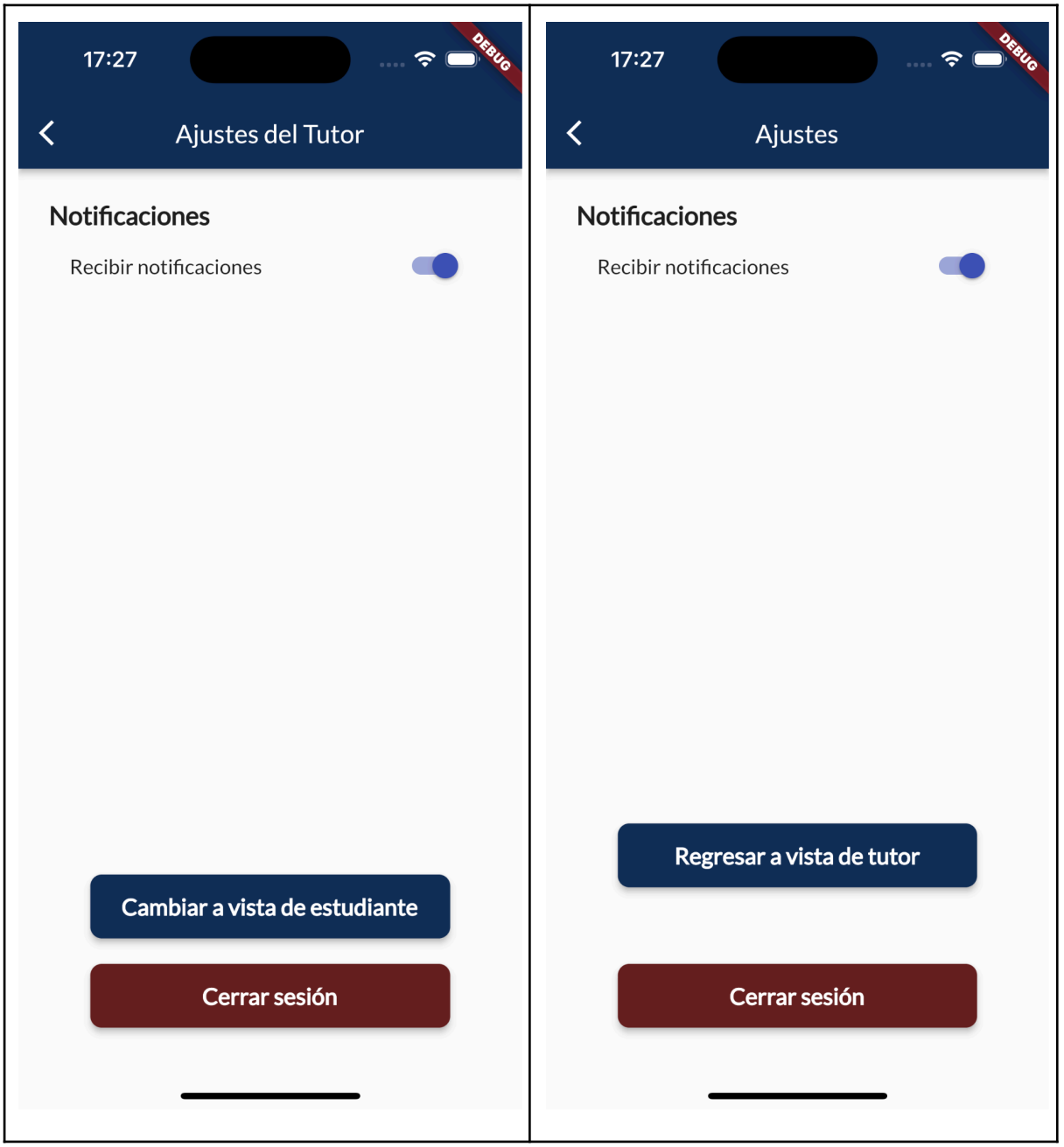
Esta pestaña permite al tutor comentar e informar sobre las tutorías que dictó. Se generan tarjetas según la cantidad de reportes pendientes que tenga el tutor, con detalles iniciales antes de navegar a la pestaña del informe.

El tutor agrega dos comentarios: uno sobre los temas que se dictaron y luego comentarios generales sobre la clase.



Perfil

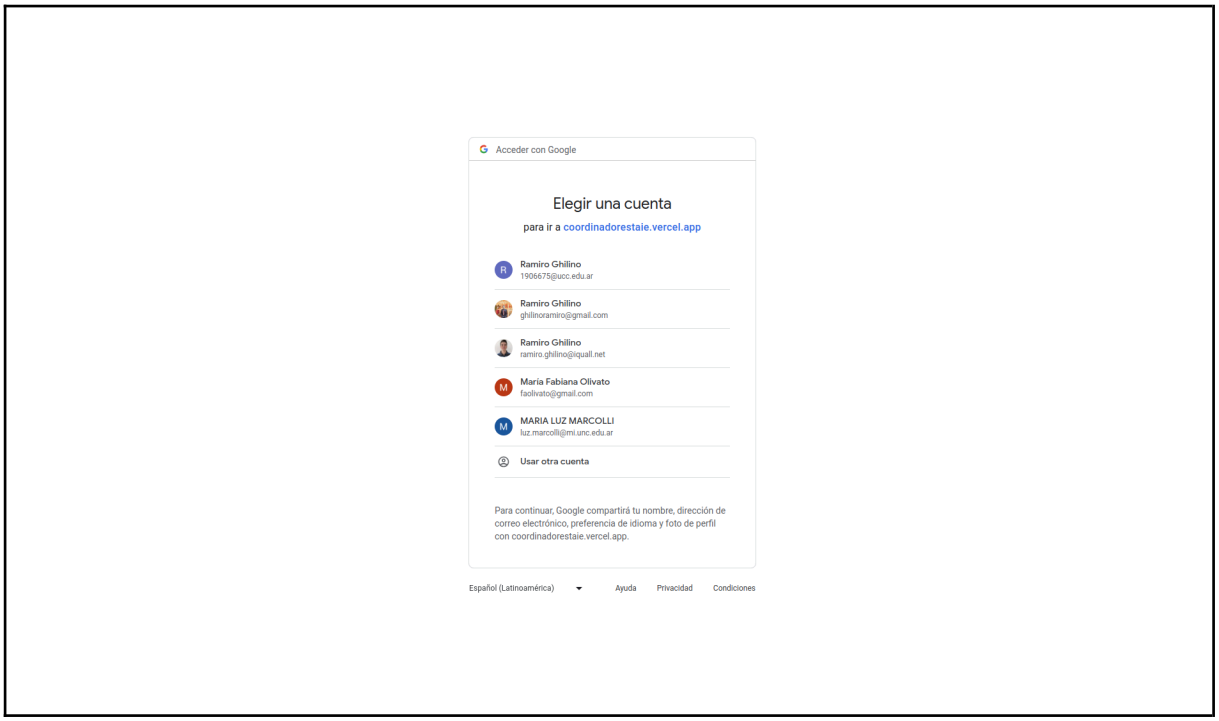
El tutor puede acceder al perfil para cambiar de rol y ver e interactuar con la aplicación como un estudiante, ya que también posee ese rol. Es posible que un tutor de segundo año necesite una clase con uno de tercer o cuarto año. Para ello, necesita presionar el botón "Cambiar a vista de estudiante". Posteriormente, puede acceder al perfil desde el modo estudiante para volver al modo tutor.



Tablero web “Tablero TAIE”

Inicio de sesión

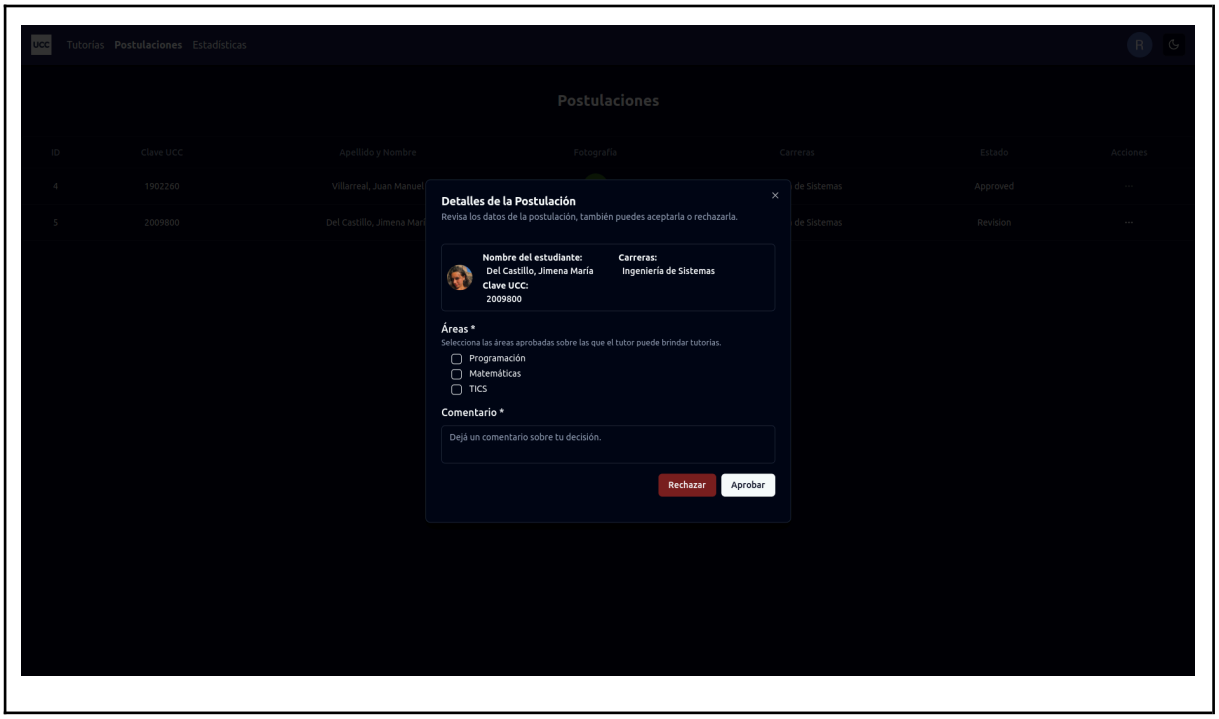
Esta pantalla recibe al coordinador al momento de abrir la página web, señalando el inicio sesión con Google. Allí se le solicitará utilizar una cuenta institucional para iniciar sesión, ya que volverá a la página principal en el caso intente utilizar otra cuenta.



Pestaña POSTULACIONES

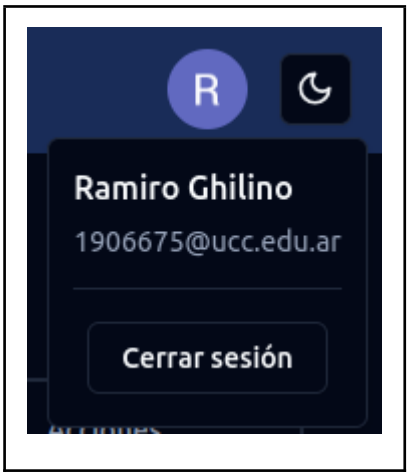
Tan pronto como inicia sesión, el usuario coordinador es redirigido a la pestaña de postulaciones, donde se muestra una tabla con los distintos tutores que están en algún momento del proceso de postulación. Esta tabla proporciona toda la información del tutor, como nombre y apellido, clave, carrera y estado.

Al presionar en 'Acciones' al lado de un tutor, se abre un diálogo con los detalles de la postulación. Allí debe indicar las áreas aprobadas y escribir un comentario sobre la decisión.



Si un usuario coordinador intenta aprobar la postulación sin indicar al menos 1 área y/o sin completar el comentario, la página muestra en el formulario, en color rojo, que campos obligatorios no han sido completados. La página tiene implementado un sistema de paginación en caso de que la cantidad de postulantes supere los 10.

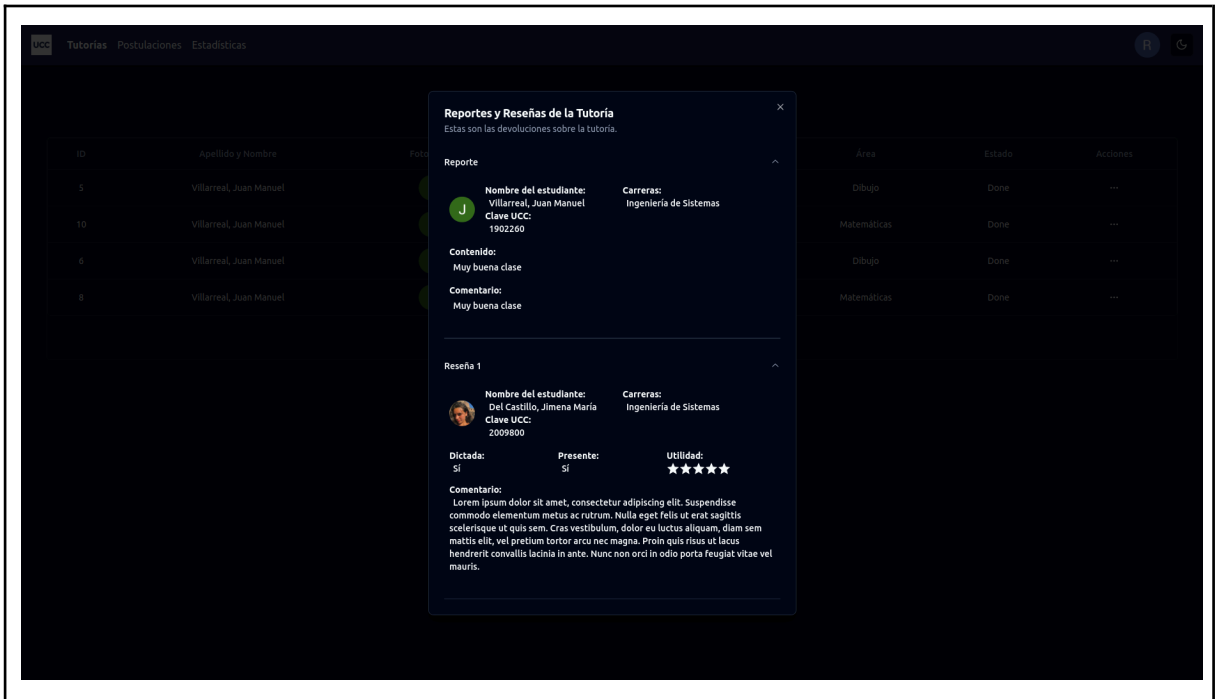
Es posible navegar a las otras dos pestañas utilizando la barra de navegación ubicada en la esquina superior izquierda. La foto de perfil, ubicada en la esquina superior derecha, también funciona como un botón que permite cerrar sesión.



Pestaña TUTORÍAS

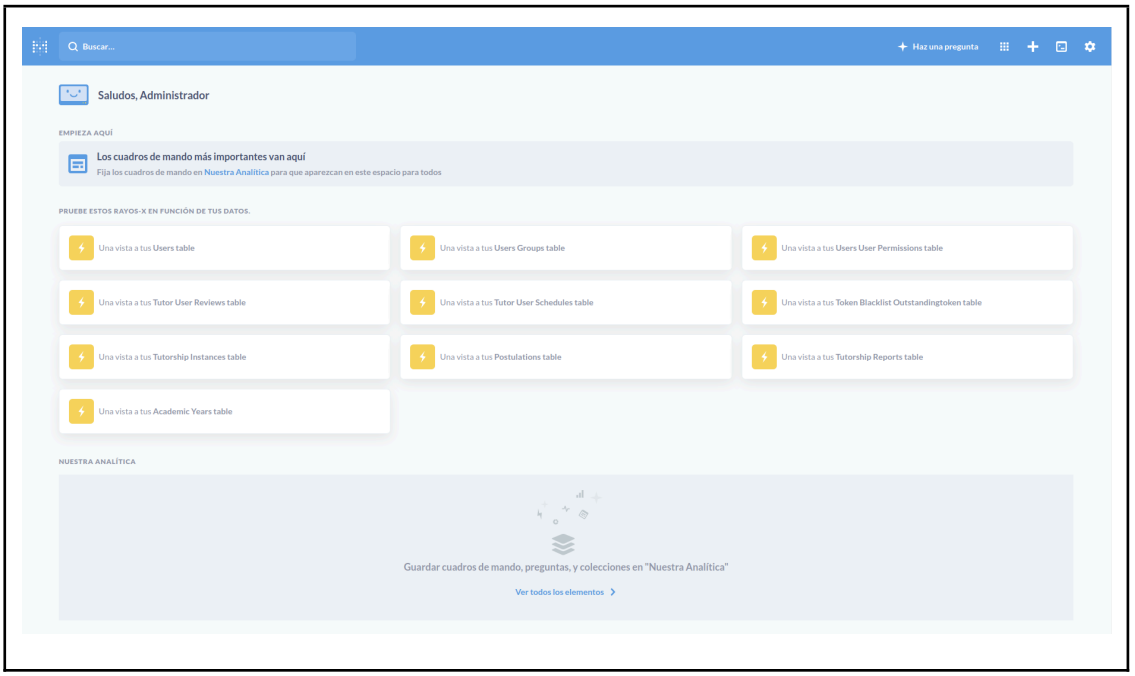
En la pestaña tutorías, el usuario coordinador ve una tabla con un listado de instancias de tutoría. Esta tabla informa el apellido y nombre del tutor que dictó esa clase, la modalidad de la instancia, la fecha y el área de conocimiento en la que se basó la consulta.

Presionando en acciones, es posible acceder tanto a los reportes por parte del tutor que dictó la clase como a la reseña por parte de los estudiantes que asistieron a la misma.



Pestaña ESTADÍSTICAS

Al presionar en este ítem, se abre una nueva pestaña que redirige al sitio de Metabase Taie, donde se aloja una instancia de la herramienta metabase relacionada con las variables y datos del sistema TAIE. Luego de iniciar sesión, es posible realizar consultas inteligentes que permiten visualizar y analizar los datos relevantes del programa.



Conclusión

La finalización de este proyecto representa el resultado de nuestro trabajo y dedicación a lo largo de 12 meses de planificación, desarrollo e implementación. Durante este tiempo, pusimos en práctica una gran parte de los conocimientos adquiridos a lo largo de nuestra carrera de Ingeniería para identificar un problema específico y diseñar una solución integral de software que aborda de manera efectiva cada uno de los desafíos presentados en el contexto del mundo real.

A pesar de haber interactuado previamente con el desarrollo de pequeños programas, reconocemos que enfrentarnos a la creación de un sistema completo fue un desafío completamente nuevo. Diseñar un sistema compuesto por distintas partes y colaborar en su desarrollo como dos desarrolladores trabajando en módulos separados no fue una meta sencilla. Ese no fue el único desafío, ya que para desarrollar los módulos clave de nuestro sistema fue necesario adentrarnos en nuevas tecnologías sobre las cuales teníamos una experiencia limitada. Esto añadió un nivel adicional de complejidad. Sin embargo, gracias a

la paciencia, la perseverancia y los conocimientos adquiridos a lo largo de nuestra formación académica, hemos logrado completar el proyecto con éxito.

Durante el proceso de desarrollo del sistema de ‘Turnero TAIE’ no solo hemos podido aplicar conocimientos adquiridos en el entorno práctico, sino que también nos ha brindado la oportunidad de aprender y crecer como profesionales en el campo de la Ingeniería en Sistemas, al interactuar con los requisitos de un cliente y generar un sistema de principio a fin, completamente utilizable si así se quisiera.

Este proyecto nos ha enseñado la importancia de la colaboración, la resolución de problemas y la adaptabilidad, habilidades fundamentales que sin duda aplicaremos en el entorno profesional. También reconocemos la importancia del esfuerzo creativo que permite generar interfaces de usuario intuitivas y agradables.

A medida que concluimos este proyecto, nos sentimos orgullosos del trabajo realizado y emocionados por el impacto potencial que nuestro sistema puede tener en la universidad y el Programa de Tutorías de Acompañamiento Integral a Estudiantes (TAIE).

Bibliografía

Almeida, J. (s.f.). Patrón MVC: Arquitectura cliente vs servidor. Recuperado de https://www.arquitecturajava.com/patron-mvc-arquitectura-cliente-vs-servidor/

Amazon Web Services. (s.f.). ¿Qué es Django? Recuperado de https://aws.amazon.com/es/what-is/django/

Amazon Web Services. (s.f.). La diferencia entre la arquitectura monolítica y de microservicios. Recuperado de https://aws.amazon.com/es/compare/the-difference-between-monolithic-and-microservices-architecture/

Apple Developer. (s.f.). Swift. Recuperado de https://developer.apple.com/swift/

Calendly. (s.f.). Education Solutions. Recuperado de https://calendly.com/es/solutions/education/

Codster. (s.f.). Desarrollo de aplicaciones web vs. escritorio. Recuperado de https://codster.io/blog/desarrollo-de-aplicaciones-web-vs-escritorio

DigitalOcean. (s.f.). Recuperado de https://www.digitalocean.com/

Django Software Foundation. (s.f.). Visión general de Django. Recuperado de https://www.djangoproject.com/start/overview/

Express.js. (s.f.). Recuperado de https://expressjs.com/

Flutter. (s.f.). Recuperado de https://flutter.dev/

Flutter. (s.f.). Desarrollo multiplataforma. Recuperado de https://flutter.dev/multi-platform

Git SCM. (s.f.). Git Documentation. Recuperado de https://git-scm.com/

JetBrains. (2023, noviembre). Django vs Flask: ¿Cuál es el mejor framework web de Python? [Entrada de blog]. Recuperado de

https://blog.jetbrains.com/pycharm/2023/11/django-vs-flask-which-is-the-best-python-web-framework/

Kinsta. (s.f.). GitLab vs GitHub: What's the Difference? Recuperado de https://kinsta.com/blog/gitlab-vs-github

Kinsta. (s.f.). PostgreSQL vs MySQL: Which is Right for You? Recuperado de https://kinsta.com/blog/postgresql-vs-mysql

Kotlin. (s.f.). Android Overview. Recuperado de https://kotlinlang.org/docs/android-overview.html

Microsoft Learn. (s.f.). ¿Qué es Xamarin? Recuperado de https://learn.microsoft.com/es-mx/xamarin/get-started/what-is-xamarin

Mobitouch. (s.f.). Aplicación móvil, aplicación web, aplicación de escritorio: conozca la diferencia. Recuperado de https://mobitouch.net/blog/mobile-app-web-app-desktop-app-know-the-difference

Red Hat. (s.f.). SOAP, REST, GraphQL y gRPC: comparación de APIs. Recuperado de https://www.redhat.com/architect/apis-soap-rest-graphql-grpc

Redalyc. (s.f.). Artículo de Revista. Recuperado de https://www.redalyc.org/articulo.oa?id=132/13228259009

Render. (s.f.). Recuperado de https://render.com/

Render. (s.f.). Plan gratuito. Recuperado de https://docs.render.com/free

React.dev. (s.f.). Recuperado de https://es.react.dev/

React Native. (s.f.). Recuperado de https://reactnative.dev/

Skooli. (s.f.). Instant Online Tutoring. Recuperado de https://www.skooli.com/

Starlette. (s.f.). Recuperado de https://www.starlette.io/

Together Platform. (s.f.). Knowledge Sharing. Recuperado de https://www.togetherplatform.com/blog/knowledge-sharing

Universidad Católica de Córdoba. (2020). Reglamento de Tutorías. Recuperado de https://www2.ucc.edu.ar/archivos/documentos/Institucional/DIGESTO/Vicerrectorado%20Academico/Secretaria%20de%20Pedagogia%20Universitaria/rr-tutorias-ucc.pdf

Universidad Católica de Córdoba. (s.f.). Secretaría de Proyección y Responsabilidad Social Universitaria. Recuperado de https://www2.ucc.edu.ar/proyeccion/secretaria-de-proyeccion-y-responsabilidad-social-universitaria/

Universidad Católica de Córdoba. (s.f.). Tutorías para alumnos. Recuperado de https://www2.ucc.edu.ar/novedades/tutorias-para-alumnos/

Universidad Católica de Córdoba. (s.f.). Unidades Académicas. Recuperado de https://www2.ucc.edu.ar/facultades/unidades-academicas/

Vercel. (s.f.). ¿Qué es Vercel? Recuperado de https://vercel.com/blog/what-is-vercel