

Desarrollo de un prototipo de un sistema para el agendamiento y gestión de tutorías académicas

Horacio Antonio Camacho Holguín

Diego Fernando Landinez León

Trabajo de grado para optar por el título de ingeniero de sistemas

Director

Gabriel Rodrigo Pedraza Ferreira

PhD. Ciencias de la computación

Codirector

Jathinson Meneses Mendoza

MSc. Ingeniería de sistemas

Universidad Industrial de Santander

Facultad de Ingenierías Físico-Mecánicas

Escuela de Ingeniería de Sistemas

Bucaramanga

2023

Dedicatoria

A mis padres, mi hermana y a mi novia, quienes han sabido entenderme y apoyarme, y a todos aquellos que de alguna forma han contribuido en mi vida hasta convertirme en quien soy.

Horacio Antonio Camacho Holguín

A mis amigos y padres, quienes han sido mi apoyo incondicional en este camino, además de ser mi inspiración y motivación constante para lograr mis metas.

Diego Fernando Landinez León

Agradecimientos

Deseamos expresar nuestro más sincero agradecimiento al profesor Gabriel Pedraza Ferreira y al profesor Jathinson Meneses Mendoza, cuya dedicación como docentes y guías ha sido invaluable en todo el proceso. Apreciamos profundamente su compromiso, conocimientos y confianza depositada en nosotros para llevar a cabo este proyecto. También queremos expresar nuestra gratitud a nuestras familias, quienes han brindado un apoyo incondicional y nos han motivado constantemente a convertirnos en mejores personas.

Tabla de contenido

Introducción	15
1. Planteamiento y justificación del problema.....	17
2. Objetivos	20
2.1 Objetivo General.....	20
2.2 Objetivos Específicos.....	20
3. Marco de referencia	21
3.1 Estado del arte.....	21
3.1.1 Preply	21
3.1.2 Coursera	21
3.2 Marco teórico	22
3.2.1 Sistemas de agendamiento	22
3.2.2 Plataforma de tutorías en línea.....	22
3.2.3 Extensibilidad	22
3.2.4 Microservicios.....	22
3.2.5 Código limpio	23
3.3 Marco tecnológico	23
3.3.1 Typescript	23
3.3.2 Nest.js.....	23
3.3.3 Dart	24
3.3.4 Flutter.....	24

3.3.5 Android studio	24
3.3.6 Git	25
3.3.7 TypeORM	25
3.3.8 PostgreSQL	25
3.1.9 TablePlus.....	26
3.3.10 Docker.....	26
3.3.11 Computación en la nube.....	26
3.3.12 SonarLint.....	26
4. Metodología	27
4.1 Marco de trabajo	27
4.2 Gestión de versiones y control de código fuente con Git.	28
5. Resultados	29
5.1 Análisis del sistema.....	29
5.1.1 Análisis del servicio actual.	29
5.1.2 Propósito del sistema, alcance del sistema, objetivos y/o criterios de éxito del proyecto. ...	30
5.1.3 Características de los usuarios	31
5.1.4 Requerimientos funcionales y no funcionales.	32
5.1.5 Casos de uso.....	35
5.1.6 Diagramas de casos de uso	46
5.1.7 Modelo de objetos.....	47
5.2 Diseño del sistema	49
5.2.1 Arquitectura del sistema.	49

5.2.2 Modelado de procesos del sistema.....	51
5.2.3 Diseño de prototipos de interfaz de usuario.....	51
5.2.4 Modelo de base de datos	52
5.3 Implementación.....	52
5.3.1 Autenticación de los usuarios.	53
5.3.2 Envío de correos	54
5.3.3 Comunicación entre los microservicios	55
5.3.4 Gestión de perfil de tutor	56
5.3.5 Solicitud de tutoría.....	58
5.3.6 Gestión de solicitudes de tutoría	62
5.3.7 Tutorías activas	64
5.3.8 Calificación de tutorías	66
5.4 Pruebas	68
5.4.1 Pruebas unitarias	69
5.4.2 Pruebas de calidad en el código	72
5.5 Documentación	72
5.5.1 Documentación de API.....	72
5.5.2 Documentación de código.....	74
5.6 Despliegue.....	75
5.6.1 Creación de imágenes Docker	75
5.6.2 Creación de instancia en el servicio EC2 de Amazon web services (AWS)	76
5.6.3 Configuración de instancia EC2	79

5.6.4 Despliegue de proyecto en instancia EC2.....	80
6. Conclusiones.....	80
7. Recomendaciones	82
Referencias Bibliográficas	83
Apéndice	86

Lista de Tablas

Tabla 1 Requerimientos funcionales de la plataforma.....	32
Tabla 2 Requerimientos no funcionales de la plataforma.....	34
Tabla 3 CU001 Registro de usuarios	35
Tabla 4 CU002 Buscar tutores.....	36
Tabla 5 CU003 Creación de perfil de tutor.....	37
Tabla 6 CU004 Realizar solicitud de tutoría.....	38
Tabla 7 CU005 Aceptar o rechazar solicitud de tutoría.....	40
Tabla 8 CU006 Enviar mensaje	42
Tabla 9 CU007 Realizar reseña y calificación de tutor	43
Tabla 10 CU008 Realizar cancelación de una tutoría.....	45

Lista de Figuras

Figura 1 Asignaturas con mayor índice de repitencia.....	15
Figura 2 Estado académico condicional UIS antes y después de la implementación del SEA	16
Figura 3 Asignaturas críticas de la escuela de ingeniería de sistemas.....	18
Figura 4 Ejemplificación de un modelo iterativo incremental.....	27
Figura 5 Ejemplificación de trunk base development.....	28
Figura 6 Modelo del servicio actual.....	29
Figura 7 Diagrama de casos de uso de estudiante.....	46
Figura 8 Diagrama de casos de uso tutor	47
Figura 9 Modelo de objetos del sistema	48
Figura 10 Arquitectura del sistema.....	50
Figura 11 Modelo de base de datos del sistema.....	52
Figura 20 Opción para activar perfil de tutor.....	56
Figura 21 Formulario de activación de perfil de tutor	57
Figura 23 Formulario de gestión de perfil	57
Figura 25 Búsqueda del tutor	58
Figura 26 Detalles del tutor.....	59
Figura 27 Formulario de solicitud de tutoría	60
Figura 30 Listado de solicitudes de tutoría.....	61
Figura 31 Detalles de solicitud	61
Figura 32 Confirmación de cancelación de tutoría.....	62

Figura 34 Solicitudes de tutoría desde perfil de tutor	63
Figura 35 Detalle de solicitud desde perfil de tutor	63
Figura 36 Confirmar rechazo de solicitud	64
Figura 37 tutorías activas de un tutor.....	65
Figura 38 Pantalla de tutoría activa	66
Figura 39 Pantalla de conversación entre estudiante y tutor.....	66
Figura 41 Listado de tutorías finalizadas	67
Figura 42 Modal de calificación de tutoría	68
Figura 43 Solicitud de tutoría sin calificar las anteriores	68
Figura 44 Ejemplo de reporte de cobertura de las pruebas.....	71
Figura 45 Cobertura de pruebas dentro del código.....	71
Figura 48 Ejemplo de colección en Postman.....	73
Figura 52 Ejemplo de interfaz para consultar el api de cada microservicio	74
Figura 57 Selección de tipo de instancia.....	77
Figura 58 Creación de claves para conectarse a la instancia	77
Figura 59 Creación de regla del grupo de seguridad	78
Figura 60 Configuración de almacenamiento instancia ec2	79
Figura 68 Prueba de petición a servicio de auth en EC2	80
Figura 69 Modelado de registro de usuarios.....	86
Figura 70 Modelado de ingreso de usuarios	86
Figura 71 Modelado de activación de perfil de tutor	86
Figura 72 Modelado de solicitud de tutoría	87

Figura 73 Modelado de consulta de tutorías	87
Figura 74 Modelado de confirmación de tutoría.....	87
Figura 75 Prototipo para pantalla de ingreso	88
Figura 76 Prototipo para pantalla de ingreso con mensaje de error.....	89
Figura 77 Prototipo para pantalla de selección de rol para ingreso	89
Figura 78 Prototipo para pantalla de primer ingreso de un usuario.....	90
Figura 79 Prototipo para pantalla de mensaje de registro.....	90
Figura 80 Prototipo de correo para activación de cuenta.....	91
Figura 81 Prototipo para consulta de tutor.....	91
Figura 82 Respuestas primera pregunta, primera encuesta.....	92
Figura 83 Segunda pregunta, primera encuesta	92
Figura 84 Tercera pregunta, primera encuesta.....	92
Figura 85 Cuarta pregunta, primera encuesta	93
Figura 86 Quinta pregunta, primera encuesta.....	93
Figura 87 Sexta pregunta, primera encuesta	93
Figura 88 Séptima pregunta, primera encuesta.....	94
Figura 89 Octava pregunta, primera encuesta.....	94
Figura 90 Novena pregunta, primera encuesta	94
Figura 91 Decima pregunta, primera encuesta	95
Figura 92 Primera pregunta, segunda encuesta.....	95
Figura 93 Segunda pregunta, segunda encuesta	95
Figura 94 Tercera pregunta, segunda encuesta	96

Lista de Apéndices

Apéndice A Modelos de procesos del sistema.....	86
Apéndice B Prototipos de interfaz	88
Apéndice C Encuestas de usuario	91

Resumen

Título: Desarrollo de un prototipo de un sistema para el agendamiento y gestión de tutorías académicas*

Autores: Horacio Antonio Camacho Holguín, Diego Fernando Landinez León**

Palabras Clave: Tutorías, Prototipo, Agendamiento, Sistema, Solicitudes de tutoría.

Descripción: Identificando las necesidades de los estudiantes de la Universidad Industrial de Santander en cuanto a las tutorías académicas y con el objetivo de reducir la deserción estudiantil, se propone el desarrollo de un prototipo de sistema de agendamiento de tutorías. El proyecto tiene como finalidad facilitar el encuentro entre estudiantes y tutores, brindando una solución integral que promueva el apoyo académico y el desarrollo de habilidades de enseñanza. El sistema de agendamiento de tutorías permitirá a los estudiantes buscar tutores en diversas áreas de estudio, así como también brindará a los estudiantes la oportunidad de ofrecerse como tutores en aquellas materias en las que tengan experiencia y conocimientos. Los estudiantes podrán acceder a perfiles de tutores y evaluar su experiencia, calificaciones y disponibilidad. A su vez, los tutores podrán recibir solicitudes de tutoría y conectar con sus estudiantes.

* Desarrollo de un prototipo de un sistema para el agendamiento y gestión de tutorías académicas

** Facultad de Ingenierías Físico-Mecánica. Escuela de Ingeniería de Sistemas e Informática.

Director: PhD. Gabriel Rodrigo Pedraza Ferreira.

Codirector: MSc. Jathinson Meneses Mendoza

Abstract

Title: Development of a prototype of a system for the scheduling and management of academic tutoring.*

Author(s): Horacio Antonio Camacho Holguín, Diego Fernando Landinez León**

Key Words: Tutoring, Prototype, Scheduling, System, Tutoring Requests.

Description: Identifying the needs of the students of the Universidad Industrial de Santander in terms of academic tutoring and with the objective of reducing student desertion, the development of a prototype of a tutoring scheduling system is proposed. The project focuses on facilitating the meeting between students and tutors, providing a comprehensive solution that promotes academic support and the development of teaching skills. The tutor scheduling system will allow students to search for tutors in various areas of study, as well as provide students with the opportunity to offer themselves as tutors in subjects in which they have experience and expertise. Students will be able to access tutor profiles and evaluate their experience, qualifications, and availability. Tutors will also be able to receive tutoring requests and connect with their students.

* Development of a prototype of a system for the scheduling and management of academic tutoring.

** School of Physical-Mechanical Engineering. Department of Systems Engineering and Informatics.

Director: PhD. Gabriel Rodrigo Pedraza Ferreira



Codirector: MSc. Jathinson Meneses Mendoza

Introducción


La Universidad Industrial de Santander, en adelante UIS, según la plataforma Top Universities, para el año 2023 se encuentra en el duodécimo puesto de universidades a nivel colombiano (QS TopUniversities, 2022). Este logro se debe en parte al alto nivel de exigencia que la universidad aplica en sus estudiantes. Sin embargo, se ha notado que muchos estudiantes han tenido dificultades en asignaturas iniciales (Figura 1) debido a la falta de fundamentación de la educación secundaria u otros factores. Para apoyar a los estudiantes en estas dificultades se ha creado el sistema de excelencia académica SEA.

Figura 1 Asignaturas con mayor índice de repitencia

**REPITENCIA POR 3 Y 4 VEZ
POR ASIGNATURA**

ASIGNATURA	MATRICULADOS		ENTERCERADOS		ENCUARTADOS		TOTAL	
	#	%	#	%	#	%	#	%
20255 - ECUACIONES DIFERENCIALES	975		182	18,67%	0	0,00%	182	18,67%
20253 - CALCULO II	959		146	15,22%	0	0,00%	146	15,22%
20252 - CALCULO I	1354		141	10,41%	0	0,00%	141	10,41%
20254 - CALCULO III	846		140	16,55%	0	0,00%	140	16,55%
22979 - ALGEBRA LINEAL I	1120		54	4,82%	0	0,00%	54	4,82%
23332 - TRATAMIENTO DE SEÑALES	187		53	28,34%	0	0,00%	53	28,34%
23834 - MECANICA DE SOLIDOS	136		46	33,82%	0	0,00%	46	33,82%
23022 - RESISTENCIA DE MATERIALES	178		41	23,03%	1	0,56%	42	23,60%
27134 - CIRCUITOS ELECTRICOS II	164		38	23,17%	0	0,00%	38	23,17%
23330 - TEORIA ELECTROMAGNETICA	258		34	13,18%	0	0,00%	34	13,18%
20113 - MECANICA ANALITICA	227		33	14,54%	0	0,00%	33	14,54%
21619 - CIRCUITOS ELECTRICOS I	157		31	19,75%	0	0,00%	31	19,75%
23021 - DINAMICA	116		29	25,00%	0	0,00%	29	25,00%
24710 - ANALISIS QUIMICO II	85		26	30,59%	0	0,00%	26	30,59%
23272 - ALGEBRA LINEAL II	198		25	12,63%	0	0,00%	25	12,63%
23039 - POTENCIA FLUIDA	128		24	18,75%	0	0,00%	24	18,75%
23018 - ESTATICA	201		23	11,44%	0	0,00%	23	11,44%
22949 - QUIMICA BASICA	601		21	3,49%	0	0,00%	21	3,49%
27124 - TECNICAS DIGITALES Y ANALOGICAS	83		20	24,10%	0	0,00%	20	24,10%



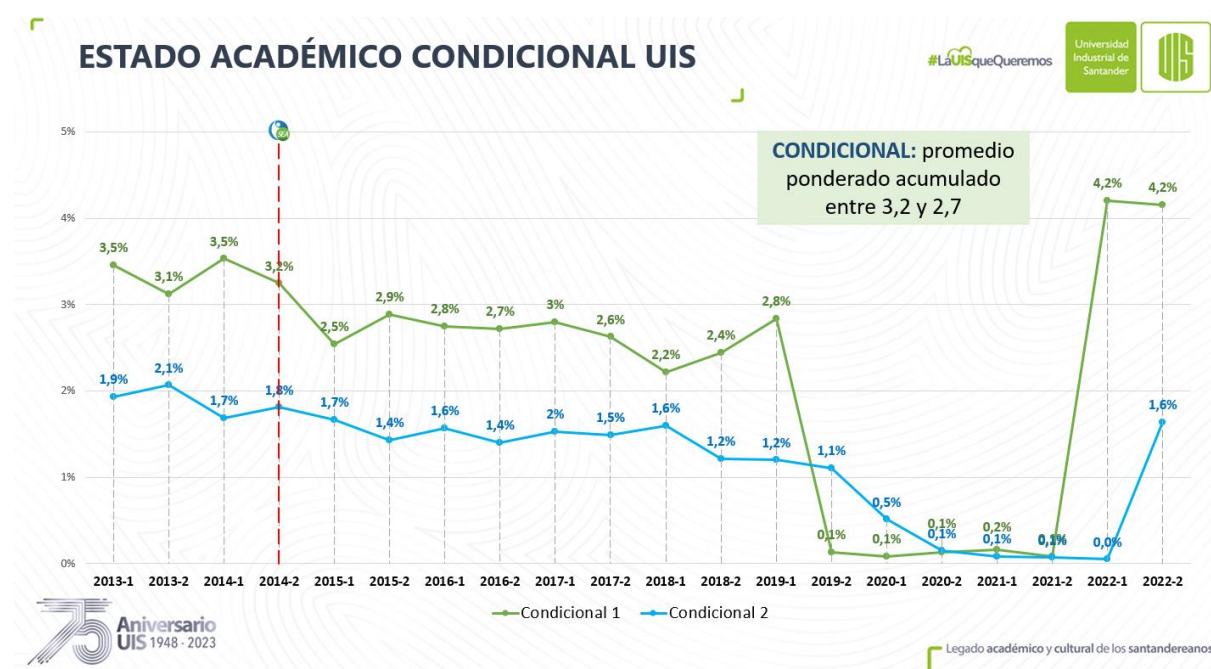


Nota: Tomado de Informe de balance académico SEA 2022-2

El sistema recolecta la información de los promedios por asignatura e informa a los directores de cada programa académico sobre los estudiantes y su desempeño respecto a cada materia del pènsum, mediante estos datos los directores de escuela pueden elegir las materias que consideren que necesitan un tutor para aumentar al nivel satisfactorio la tasa de aprobación

Aunque este programa ha ayudado a muchos estudiantes, se ha evidenciado que no ha sido suficiente para resolver por completo las dificultades en el aprendizaje de los estudiantes, especialmente en los semestres posteriores a la pandemia (Figura 2). Por tanto, es necesario crear otras opciones o alternativas que permitan mejorar la condición de los estudiantes.

Figura 2 Estado académico condicional UIS antes y después de la implementación del SEA



Nota: Tomado de informe de balance académico SEA 2022-2

1. Planteamiento y justificación del problema

Las tutorías son una herramienta con un gran valor, ya que permite al tutor compartir sus conocimientos en sus áreas de experticia, mientras que permite al estudiante resolver las dudas que por motivos varios no pudieron ser resueltas en la clase con el profesor, adicionalmente esta ayuda puede llegar a ser personalizada, a diferencia de un salón de clase normal en el que el profesor debe seguir avanzando en tema por lo que no puede dedicarle una gran mayoría del tiempo a resolver dudas.

La tutoría tiene como beneficio además de aclararle dudas al estudiante, también le permite al tutor recibir una compensación por sus conocimientos, incitando a más estudiantes que poseen habilidades en ciertas áreas a inscribirse en el programa, lo que proporcionaría un aumento en la cantidad de tutores de calidad, permitiendo así que los estudiantes puedan entender con facilidad las materias más complicadas, disminuyendo la deserción o la expulsión de la universidad.

Desafortunadamente, se ha observado que el programa implementado hasta ahora no ha sido suficiente para dar solución a la problemática en la escuela de ingeniería de sistemas. Una de las posibles razones es que solo existe un tutor por cada materia elegida por el director de escuela, lo que elimina completamente la competencia entre tutores.

Figura 3 Asignaturas críticas de la escuela de ingeniería de sistemas



Nota: Informe de balance académico SEA 2022-2

Es importante tener en cuenta que el SEA no cubre todas las materias, lo que aumenta la problemática para los estudiantes en caso de tener preguntas en asignaturas no atendidas por esta entidad. Esto puede llevar a la insatisfacción de sus necesidades de aprendizaje en su proceso de formación, como el caso de la asignatura sistemas de información, la cual solo posee un 39% de aprobación y no fue considerada para una asignación de tutorías.

De la misma manera que los estudiantes pueden llegar a tener dudas en áreas que no tienen tutorías disponibles, los tutores también pueden llegar a tener una gran habilidad en aquellas áreas que fueron consideradas con “suficiente porcentaje de aprobación” por lo que no tienen una plataforma o un sistema que les permita darse a conocer y poder compartir este conocimiento de forma remunerada.

Con el propósito de dar solución a la problemática mencionada, en este proyecto se desarrollará un sistema de agendamiento de tutorías en línea, el cual le permitirá a los tutores conectar con estudiantes que necesiten apoyo, y les ofrecerá a los estudiantes la posibilidad de agendar horarios de tutoría con su tutor preferido.

Para proveer de un servicio de calidad se analizarán las necesidades y expectativas de los usuarios (estudiantes y tutores) con el objetivo de definir los requisitos funcionales y no funcionales del sistema, además de un diseño de la interfaz de usuario y el desarrollo un sistema de reseñas, calificación y validación de estas, para garantizar que los estudiantes siempre encuentren un tutor de calidad, y puedan tomar decisiones informadas, y también incluirá un sistema de chat entre tutor y estudiantes para facilitar la comunicación y evitar la necesidad de compartir información personal, como el número de teléfono, para usuarios que no deseen hacerlo.

El sistema mencionado se espera que sea útil tanto para estudiantes como para los tutores y permita mejorar la experiencia del aprendizaje, además de aumentar los porcentajes de aprobación de todas las materias, no solo las más críticas, a diferencia del sistema actualmente utilizado.

Pregunta de investigación: ¿De qué forma es posible mejorar la eficacia de un sistema de tutorías de la Universidad Industrial de Santander y establecer una relación de confianza entre estudiantes y tutores?

2. Objetivos

2.1 Objetivo General

Diseñar un sistema software de relacionamiento para tutorías universitarias que permita a los estudiantes ofrecer y solicitar servicios de tutorías según sus habilidades y necesidades.

2.2 Objetivos Específicos

- Realizar encuestas para, a través del análisis de las necesidades y expectativas de los usuarios del sistema (estudiantes y tutores), definir los requisitos funcionales y no funcionales.
- Realizar el diseño arquitectural y de detalle de la plataforma de relacionamiento entre tutores y estudiantes.
- Implementar un prototipo de la plataforma de relacionamiento para tutorías universitarias en el cual se incluirán:
 - Módulo de agendamiento de tutorías.
 - Módulo de calificación y reseña de tutores.
 - Sección de chat entre estudiantes y tutores.
- Validar el prototipo a través de un plan de pruebas de calidad.

3. Marco de referencia

3.1 Estado del arte

3.1.1 Preply

Preply es una plataforma educativa en línea que conecta profesores y alumnos de todo el mundo. Ofrece cursos de idiomas y clases particulares de varias materias. A diferencia de Preply, este proyecto está dirigido específicamente a estudiantes de la UIS que desean obtener ingresos adicionales y recibir ayuda académica (Preply, s.f).

3.1.2 Coursera

Coursera es una plataforma de educación virtual que ofrece cursos de muy diversos temas en distintos idiomas, con la finalidad de brindar acceso a clases de educación superior a todos los usuarios interesados. Los usuarios tienen la posibilidad de aprender nuevas habilidades laborales, completar programas especializados y mucho más (Aprender Gratis, 2022, sección "¿Qué es Coursera?").

3.1.3 Wyzant

Wyzant es un mercado en línea para ayudar a los estudiantes a conseguir tutores profesionales para un ambiente de uno a uno. La compañía también ofrece una aplicación móvil para facilitar la conexión entre los tutores y los estudiantes. (Gigonway, 2020, sección "Tutoría Wyzant").

3.2 Marco teórico

3.2.1 Sistemas de agendamiento

Un sistema de agendamiento es un software de gestión diseñado para simplificar y optimizar los procesos de reserva de citas, es una herramienta que permite a los usuarios coordinar y programar citas o reuniones con una persona o entidad. (Reservo, 2022, sección "¿Por qué un sistema de agendamiento ayuda a tu empresa?").

3.2.2 Plataforma de tutorías en línea

Las plataformas de tutorías en línea son herramientas que permiten a los estudiantes encontrar y contratar a tutores para recibir tutorías de manera remota, ya sea a través de videollamadas o chat en línea. Estas plataformas suelen incluir un directorio de tutores con sus perfiles y especialidades, así como un sistema de reserva y pago en línea. es.

3.2.3 Extensibilidad

La extensibilidad es la capacidad de una aplicación o sistema para adaptarse y expandirse fácilmente con el objetivo de satisfacer las necesidades de negocio cambiantes sin tener que reescribir completamente el código o la arquitectura existente.

3.2.4 Microservicios

Los microservicios son una arquitectura de software la cual consiste en dividir una aplicación en componentes o servicios más pequeños y manejables, cada uno de los cuales se centra en una tarea específica. Cada servicio tiene una función específica y se comunican entre sí

a través de interfaces estandarizadas para ofrecer la funcionalidad total de la aplicación (Mediacloud, 2023).

3.2.5 Código limpio

El código limpio es un término utilizado para describir código que es simple, fácil de leer y entender. El código limpio se adhiere a ciertos principios y buenas prácticas de diseño y estilo de programación, y evita el uso de trucos o técnicas de programación que puedan hacer que el código sea difícil de entender o mantener. (Hostgator, 2022, sección "Clean Code: Código limpio, ¿qué es?").

3.3 Marco tecnológico

3.3.1 Typescript

Typescript es un lenguaje de programación libre y de código abierto, que posee un tipado estricto que se basa en JavaScript y se emplea principalmente para desarrollar aplicaciones de JavaScript a gran escala. (ITDO, 2021, sección "¿Qué es TypeScript, y por qué utilizarlo?").

3.3.2 Nest.js

Nest.js es una estructura para el desarrollo de aplicaciones eficientes en el servidor para Node.js. Es un marco de alto nivel que se basa en el patrón de arquitectura de aplicaciones basadas en el servidor MVC (Modelo-Vista-Controlador) y está diseñado para ser fácil de usar y escalable, ya que combina elementos de programación orientada a objetos, programación

imperativa y programación funcional (Epitech, 2022, sección "NestJS: ¿Qué es y por qué es tan interesante este framework Node.Js?").

3.3.3 Dart

Dart es un lenguaje de programación de código abierto desarrollado por Google. Se diseñó para ser rápido, fácil de emplear y altamente escalable. Dart se puede disponer para desarrollar aplicaciones para la web, para dispositivos móviles y para servidores (Back4app, s.f., sección "Flutter vs Dart: ¿cuál es mejor?").

3.3.4 Flutter

Flutter es un kit de herramientas de desarrollo de aplicaciones móviles desarrollado por Google. Se basa en el lenguaje de programación Dart y permite a los desarrolladores crear aplicaciones móviles nativas para Android y iOS a partir del mismo código fuente (Back4app, s.f., sección "Flutter vs Dart: ¿cuál es mejor?").

3.3.5 Android studio

Android Studio es un entorno de desarrollo integrado (IDE) para el sistema operativo Android. Es el entorno de desarrollo oficial de Google para el desarrollo de aplicaciones móviles para Android (Developers, s.f., sección "Introducción a Android Studio").

3.3.6 Git

Git es un sistema de control de versiones de código fuente. Es una herramienta muy útil para los desarrolladores de software, ya que permite llevar un registro de todos los cambios realizados en el código fuente de un proyecto y facilitar la colaboración entre múltiples desarrolladores. (Hostinger, 2023, sección "Qué es GitHub y cómo empezar a usarlo").

3.3.7 TypeORM

TypeORM es una herramienta de mapeo de objetos relacional (ORM, por sus siglas en inglés) de código abierto para Node.js y Typescript. Le permite definir entidades (es decir, objetos que representan filas en una tabla de base de datos) y realizar operaciones de base de datos, como consultar, insertar, actualizar y eliminar datos, de manera sencilla empleando código JavaScript o Typescript.

3.3.8 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacionales de código abierto. Es uno de los sistemas de bases de datos más populares y usados en el mundo, con una gran cantidad de características avanzadas y una comunidad activa de desarrolladores y usuarios (OVHcloud, s.f., sección "¿Qué es PostgreSQL?").

3.1.9 TablePlus

TablePlus es una herramienta de gestión de bases de datos y SQL que se utiliza para conectarse a bases de datos locales o remotas, efectuar consultas SQL y visualizar y editar los datos de las tablas. (Capterra, s.f., sección "¿Qué es TablePlus?").

3.3.10 Docker

Docker es una plataforma de contenedores de código abierto que permite a los desarrolladores empaquetar y distribuir aplicaciones junto con todas sus dependencias en un contenedor. Esto hace que sea más fácil desplegar y ejecutar aplicaciones en cualquier entorno, ya sea en un servidor local o en la nube. (ComputerWeekly, s.f., sección "Docker").

3.3.11 Computación en la nube

La computación en la nube es un modelo de informática que permite a las empresas acceder a recursos de informática, como servidores, almacenamiento en línea y aplicaciones, a través de Internet. En lugar de tener que comprar y mantener físicamente estos recursos, las empresas pueden usar la informática en la nube para acceder a ellos de manera más eficiente y flexible (IaaS), plataforma (PaaS) o software (SaaS).

3.3.12 SonarLint

SonarLint es una herramienta que permite detectar problemas en el código fuente de una aplicación y para ayudar a garantizar que el código cumpla con los estándares de calidad y estilo establecidos.

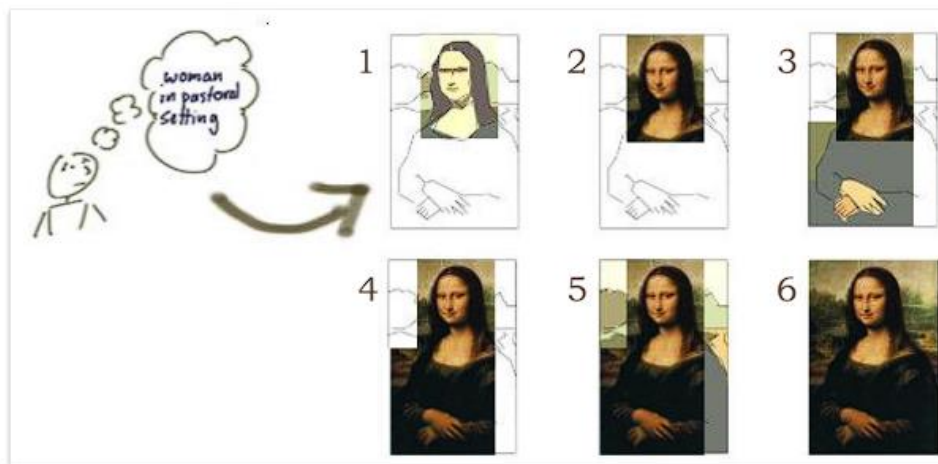
4. Metodología

4.1 Marco de trabajo

Para el desarrollo de este proyecto se trabajará bajo SCRUM, un marco de trabajo ágil que busca obtener resultados rápidos (prototipos) a través de un enfoque flexible y un ciclo de vida iterativo e incremental. Con esta metodología, se pretende garantizar que el proyecto cumpla con los requisitos establecidos y evitar posibles errores que puedan generar reprocesos.

Para asegurar el éxito del proyecto, el desarrollo será dividido en las siguientes etapas: análisis del sistema, definición de requerimientos, diseño, implementación, pruebas y documentación. Cada una de estas etapas será realizada de manera iterativa e incremental, permitiendo avanzar en el proyecto de manera eficiente y garantizando la calidad del resultado final.

Figura 4 Ejemplificación de un modelo iterativo incremental



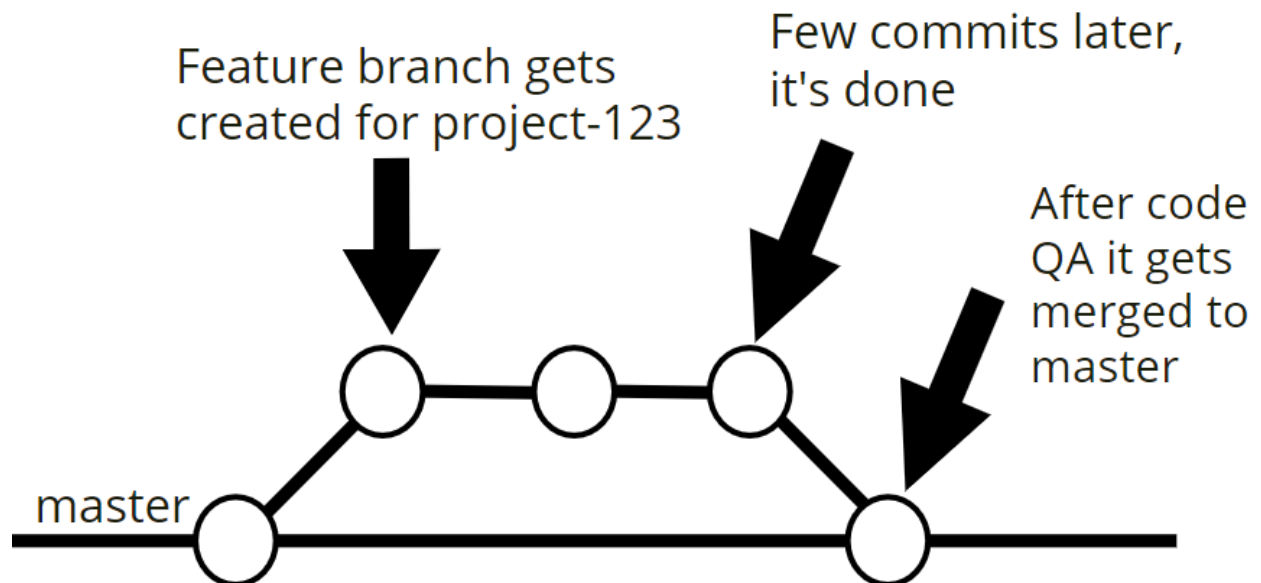
Nota: Tomado de Revisiting the Iterative Incremental Mona Lisa, Steven Thomas

4.2 Gestión de versiones y control de código fuente con Git.

Para la gestión del código fuente de este proyecto se utilizará la metodología Trunk Based, una metodología que se adapta bien en equipos de desarrollo pequeño como en este caso en que el proyecto es desarrollado por un equipo de dos personas.

Esta metodología consiste en mantener una rama principal (trunk) en la que se integran todos los cambios realizados por los desarrolladores. Entre las ventajas de Trunk Based se encuentra la facilidad para mantener un flujo constante de integración continua, lo que permite una entrega rápida de nuevas funcionalidades y corrección de errores de forma más eficiente.

Figura 5 Ejemplificación de trunk base development



Nota: Tomado de <https://www.freecodecamp.org/news/what-is-trunk-based-development/>

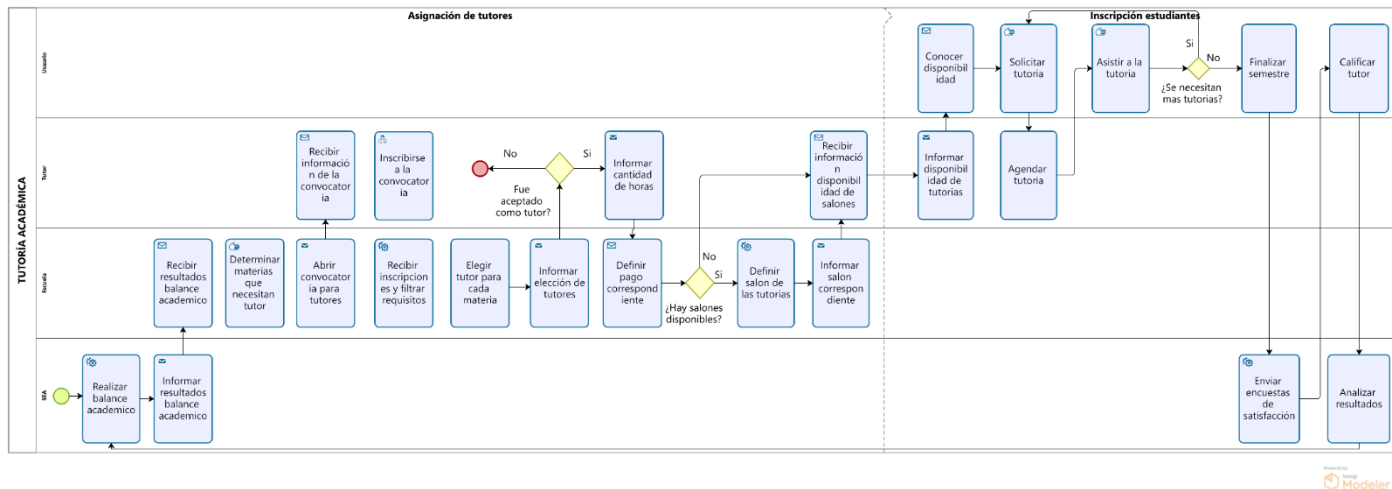
5. Resultados

5.1 Análisis del sistema

5.1.1 Análisis del servicio actual.

Antes de implementar el proyecto de sistema de agendamiento de tutorías, es importante comprender cómo funciona el actual servicio de tutorías en la universidad. A continuación, se presenta un diagrama que muestra el proceso y las interacciones involucradas en las tutorías.

Figura 6 Modelo del servicio actual



Fuente: Autores bajo guía de administrativos del decanato de físicomecánicas

5.1.2 Propósito del sistema, alcance del sistema, objetivos y/o criterios de éxito del proyecto.

El propósito del sistema es facilitar una alternativa para tutorías universitarias en la Universidad Industrial de Santander (UIS) mediante la creación de una plataforma en línea que permita a los estudiantes y tutores ofrecer y solicitar servicios de tutorías según sus habilidades y necesidades.

El sistema permitirá a los estudiantes y tutores de la UIS registrarse y crear perfiles en la plataforma. Por parte del tutor, podrá especificar sus habilidades, materias y horarios disponibles. Los estudiantes podrán buscar tutores según la materia que necesiten reforzar y las habilidades del tutor. Los tutores podrán aceptar o rechazar solicitudes de tutoría y conectar con sus estudiantes. La plataforma también contará con una sección de chat en la que los tutores y estudiantes podrán comunicarse antes y durante la tutoría.

El objetivo principal del proyecto es diseñar, implementar y validar un sistema software de relacionamiento para tutorías universitarias en la UIS. Los objetivos específicos incluyen realizar encuestas para definir los requisitos funcionales y no funcionales, diseñar la arquitectura y los detalles de la plataforma, implementar un prototipo funcional y validar el sistema a través de un plan de pruebas de calidad. El criterio de éxito del proyecto será la aceptación y adopción por parte de los estudiantes y tutores de la UIS, así como la mejora de la eficiencia y calidad de las tutorías universitarias.

Este proyecto se considera finalizado, una vez se cumplan los siguientes aspectos:

- Se ha desarrollado el prototipo en base a los requerimientos funcionales y no funcionales surgidos de las encuestas y análisis del sistema de tutorías.
- Se han realizado las pruebas para validar el prototipo.
- Se ha realizado la documentación del desarrollo.
- El sistema cumple con los objetivos y criterios establecidos.

5.1.3 Características de los usuarios

A continuación, se presentan las características de los usuarios que interactúan con el sistema, es importante aclarar que los estudiantes de la universidad industrial de Santander pueden registrarse en la plataforma y ejercer el rol de tutores y estudiantes que solicitan una tutoría:

- **Estudiante:** Como estudiante, un usuario puede buscar tutores para una materia específica, ver las calificaciones y reseñas de un tutor, solicitar una tutoría, comunicarse con el tutor por medio del módulo chat y finalmente calificar esa tutoría una vez finalizada.
- **Tutor:** Como tutor, un usuario puede ofrecer tutorías de una o más materias, definir sus horarios de disponibilidad y aceptar o rechazar las solicitudes de tutoría de los estudiantes.
- **Administrador:** Como administrador, el usuario en este prototipo puede bloquear y desbloquear el acceso de un usuario a la plataforma.

5.1.4 Requerimientos funcionales y no funcionales.

Después de analizar el servicio de tutorías que ofrece la universidad y de realizar encuestas a estudiantes, se han identificado los siguientes requisitos funcionales y no funcionales que se deben considerar en el diseño de la plataforma de tutorías.

5.1.4.1 Requerimientos funcionales

La siguiente tabla presenta los requerimientos funcionales del sistema de relacionamiento para tutorías universitarias. Estos requerimientos describen las funcionalidades específicas que el sistema debe tener para cumplir con los objetivos del proyecto y satisfacer las necesidades de los usuarios.

Tabla 1 Requerimientos funcionales de la plataforma

Requerimiento	Descripción
Registro de usuarios	Los usuarios deben poder registrarse en la plataforma utilizando su correo institucional de la UIS (@correo.uis.edu.co) y completando un formulario con su información personal
Búsqueda de tutores	Los estudiantes deben poder buscar tutores según la materia que necesiten reforzar.
Registro de materias y horarios	Los tutores deben poder registrar en la plataforma las materias para las cuales pueden dictar tutorías y los horarios en los cuales están

	disponibles para ofrecer sus servicios.
Solicitud y respuesta de tutorías	Los estudiantes deben poder hacer solicitudes de tutoría a un tutor específico, indicando los detalles de la tutoría como el tema, la materia y la fecha. El tutor debe recibir la solicitud y poder aceptar o rechazar la tutoría. La plataforma debe notificar a ambos usuarios sobre la solicitud y su estado (aceptada o rechazada)
Sección de chat	La plataforma debe contar con una sección de chat en la que los tutores y estudiantes puedan comunicarse
Reseñas y calificaciones por materia	Los estudiantes deben poder dejar reseñas y calificaciones para cada materia que un tutor dicta. El tutor debe tener una calificación y reseña por cada materia que ofrece tutorías.
Cancelación de tutorías	Los usuarios, deben poder cancelar una tutoría, con el rol de tutor como de estudiante, si se encuentra dentro del plazo definido para hacerlo.

5.1.4.2 *Requerimientos no funcionales*

La siguiente tabla presenta los requerimientos no funcionales del sistema de relacionamiento para tutorías universitarias. Los requerimientos no funcionales son esenciales para garantizar que el sistema sea eficiente, seguro, escalable y fácil de usar para los usuarios.

Tabla 2 Requerimientos no funcionales de la plataforma

Requerimiento	Descripción
Seguridad	La plataforma debe contar con medidas de seguridad para proteger los datos personales y de contacto de los usuarios.
Usabilidad	La plataforma debe ser fácil de usar y navegar tanto para tutores como para estudiantes.
Disponibilidad	La plataforma debe estar disponible 99% del tiempo.
Escalabilidad	La plataforma debe ser capaz de manejar un volumen de 3000 usuarios sin disminuir su rendimiento.
Compatibilidad	La plataforma debe ser compatible para los dispositivos Android y iOS

5.1.5 Casos de uso

A continuación, se presentarán los casos de uso que se han creado a partir de los requerimientos del sistema de relacionamiento para tutorías universitarias. Estos casos de uso describen las diferentes funcionalidades del sistema desde la perspectiva de los usuarios y son esenciales para comprender el funcionamiento del sistema en su conjunto.

Tabla 3 CU001 Registro de usuarios

Nombre del caso de uso	Registrar usuario
Requerimiento	Registro de usuarios
Actores principales	Usuario
Descripción	El estudiante de la universidad industrial de Santander realiza registro en la plataforma de tutorías. Por defecto el usuario es registrado como estudiante, para ser tutor deberá crear su perfil de tutor.
Precondiciones:	El correo institucional del estudiante no debe haber sido previamente registrado en la plataforma.
Postcondiciones	
Flujo normal	1. El usuario selecciona la opción de registro.

-
2. El usuario ingresa los datos solicitados
 3. El sistema valida los datos ingresados por el usuario.
 4. La aplicación muestra el mensaje de error si lo hay.
 5. El sistema envía un correo de activación de cuenta al correo ingresado.
 6. La aplicación muestra una ventana indicando que debe autenticar su cuenta por medio del correo enviado.
 7. El usuario recibe correo de activación
 8. El usuario ingresa por enlace de activación y autentica la cuenta.
-

Tabla 4 CU002 Buscar tutores

Nombre del caso de uso	Buscar tutores
Requerimiento	Búsqueda de tutores
Actores principales	Estudiante
Descripción	El estudiante desea buscar un tutor acorde a sus necesidades.

Precondiciones:	El usuario debe estar autenticado en la aplicación.
Postcondiciones	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario ingresa con el rol de estudiante. 2. El usuario pulsa sobre la opción “buscar tutor”. 3. El usuario ingresa el nombre del tutor o materia para realizar consulta de tutor. 4. La aplicación despliega una lista de los tutores que coinciden con el criterio ingresado

Tabla 5 CU003 Creación de perfil de tutor

Nombre del caso de uso	Creación de perfil de tutor
Requerimiento	Registro de materias y horarios
Actores principales	Tutor
Descripción	El tutor necesita crear su perfil como tutor y definir las materias para las cuales desea dictar tutorías y el horario de preferencia en el que

	puede hacerlo.
Precondiciones:	El usuario debe estar autenticado en la aplicación.
Postcondiciones	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la opción “Activar perfil de tutor”. 2. La aplicación muestra pantalla con formulario para creación de perfil de tutor. 3. El usuario llena los campos solicitados, tales como descripción básica, materias y preferencias de horarios y precios para dictar las tutorías. 4. El sistema registra los datos ingresados por el usuario y le asigna rol de tutor. 5. La aplicación muestra el mensaje de error si lo hay.

Tabla 6 CU004 Realizar solicitud de tutoría

Nombre del caso de uso	Realizar solicitud de tutoría
Requerimiento	Solicitud y respuesta de tutorías

Actores principales	Estudiante
Descripción	El usuario, ingresando por el rol de estudiante, realiza la búsqueda del tutor por el módulo de consulta, selecciona un tutor y desea realizar una solicitud de tutoría.
Precondiciones:	El usuario ha ingresado con el rol de estudiante, ha consultado por la opción de buscar tutor y se encuentra en la ventana de información del tutor.
Postcondiciones	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la opción de realizar solicitud de tutoría. 2. El usuario ingresa los datos solicitados por la aplicación para la realización de la solicitud de tutoría. 3. El usuario pulsa sobre la opción de enviar solicitud. 4. El sistema registra la solicitud de tutoría. 5. La aplicación muestra el mensaje de error si lo hay. 6. El sistema notifica la solicitud de

tutoría al tutor.

Tabla 7 CU005 Aceptar o rechazar solicitud de tutoría

Nombre del caso de uso	Aceptar o rechazar solicitud de tutoría
Requerimiento	Solicitud y respuesta de tutorías
Actores principales	Tutor
Descripción	El usuario, ingresando por el rol de tutor, pulsa sobre la opción de ‘Solicitudes de tutoría’, la aplicación le muestra la lista de solicitudes de tutorías vigentes, el tutor selecciona la solicitud de interés, la lee y desea aceptar o rechazar.
Precondiciones:	El usuario se encuentra autenticado
Postcondiciones	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario ingresa con el rol de tutor. 2. El usuario pulsa sobre la opción “Solicitudes de tutorías”. 3. La aplicación muestra un listado de solicitudes de tutoría. 4. El usuario pulsa sobre una solicitud en el listado.

-
5. La aplicación muestra una ventana con detalles sobre la solicitud.
 6. El tutor selecciona entre las opciones de aceptar o rechazar.
 7. Si el tutor ha pulsado en rechazar, la aplicación le muestra una ventana con un formulario para que el tutor tenga posibilidad de informar al estudiante la razón del rechazo.
 8. El sistema tramita la decisión del tutor.
 9. La aplicación muestra el mensaje de error si lo hay.
 10. El sistema notifica al estudiante sobre la decisión del tutor.

Flujo alternativo

1. La aplicación notifica al usuario sobre solicitud de tutoría.
2. El usuario pulsa sobre notificación.
3. La aplicación muestra ventana con detalles de sobre la solicitud.
4. El tutor selecciona entre las opciones de aceptar o rechazar.
5. Si el tutor ha pulsado en rechazar, la

	<p>aplicación le muestra una ventana con un formulario para que el tutor tenga posibilidad de informar al estudiante la razón del rechazo.</p> <p>6. El sistema tramita la decisión del tutor.</p> <p>7. La aplicación muestra el mensaje de error si lo hay.</p> <p>8. El sistema notifica al estudiante sobre la decisión del tutor.</p>
--	--

Tabla 8 CU006 Enviar mensaje

Nombre del caso de uso	Enviar mensaje
Requerimiento	Sección de chat
Actores principales	Usuario
Descripción	Una vez que el tutor acepta la solicitud de tutoría, el estudiante y el tutor pueden comunicarse por medio del módulo de chat de la aplicación.
Precondiciones:	El usuario se ha autenticado en la aplicación, pulsa sobre la opción ‘tutorías activas’ y selecciona una de las tutorías en la lista

	desplegada por la aplicación y se encuentra con una ventana donde se encuentran los detalles de la tutoría.
Postcondiciones	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la opción “ir al chat”. 2. La aplicación le muestra al usuario una ventana común de chat entre estudiante y tutor. 3. El usuario escribe un mensaje y pulsa el botón de ‘enviar’ 4. El sistema envía el mensaje y notifica al otro usuario.

Tabla 9 CU007 Realizar reseña y calificación de tutor

Nombre del caso de uso	Realizar reseña y calificación de tutor
Requerimiento	Reseñas y calificaciones por materia
Actores principales	Estudiante
Descripción	Una vez realizada la tutoría, el estudiante tiene la posibilidad de calificar al tutor y de escribir una reseña sobre la tutoría.

Precondiciones: La tutoría se encuentra en estado de finalizada y el estudiante aún no ha realizado la reseña del servicio.

Postcondiciones

Flujo normal

1. El estudiante pulsa sobre la opción de historial de tutorías.
 2. La aplicación muestra una lista con las tutorías tomadas por el estudiante.
 3. El estudiante selecciona una tutoría en el historial.
 4. La aplicación muestra una ventana con información básica de la tutoría.
 5. El estudiante pulsa sobre la opción de calificar tutoría.
 6. La aplicación muestra una ventana con un formulario para ingresar la calificación y reseña de la tutoría.
 7. El estudiante ingresa los datos y pulsa sobre el botón de enviar reseña.
 8. El sistema registra la calificación y reseña.
-

Tabla 10 CU008 Realizar cancelación de una tutoría.

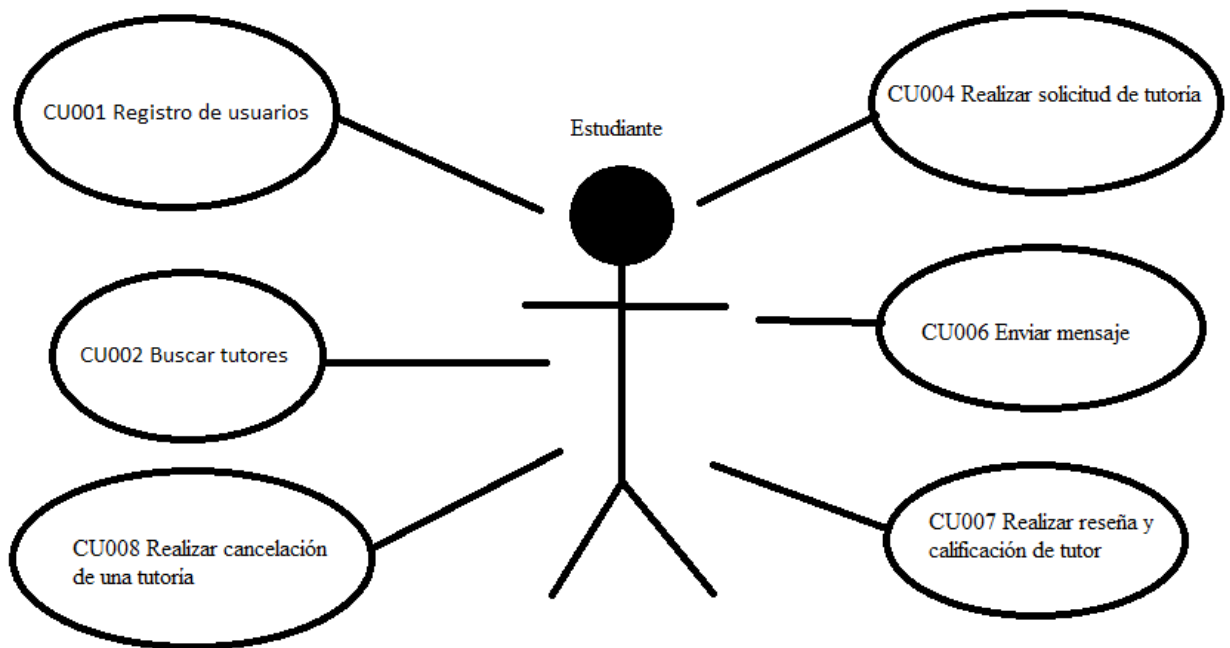
Nombre del caso de uso	Realizar cancelación de una tutoría.
Requerimiento	Cancelación de tutorías
Actores principales	Usuario
Descripción	El usuario, autenticado en la aplicación con el rol de estudiante o de tutor, desea cancelar una solicitud en estado vigente.
Precondiciones:	La tutoría debe estar en estado vigente y la cancelación debe estar dentro del plazo asignado para la cancelación de tutorías.
Postcondiciones	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la opción de ‘tutorías activas’. 2. La aplicación despliega una lista con las tutorías vigentes para el usuario. 3. El usuario selecciona la opción de cancelar tutoría. 4. La aplicación muestra una ventana de confirmación para cancelar la tutoría. 5. El usuario pulsa sobre el botón ‘cancelar’

-
6. El sistema tramita la cancelación de la tutoría y notifica al otro usuario.
-

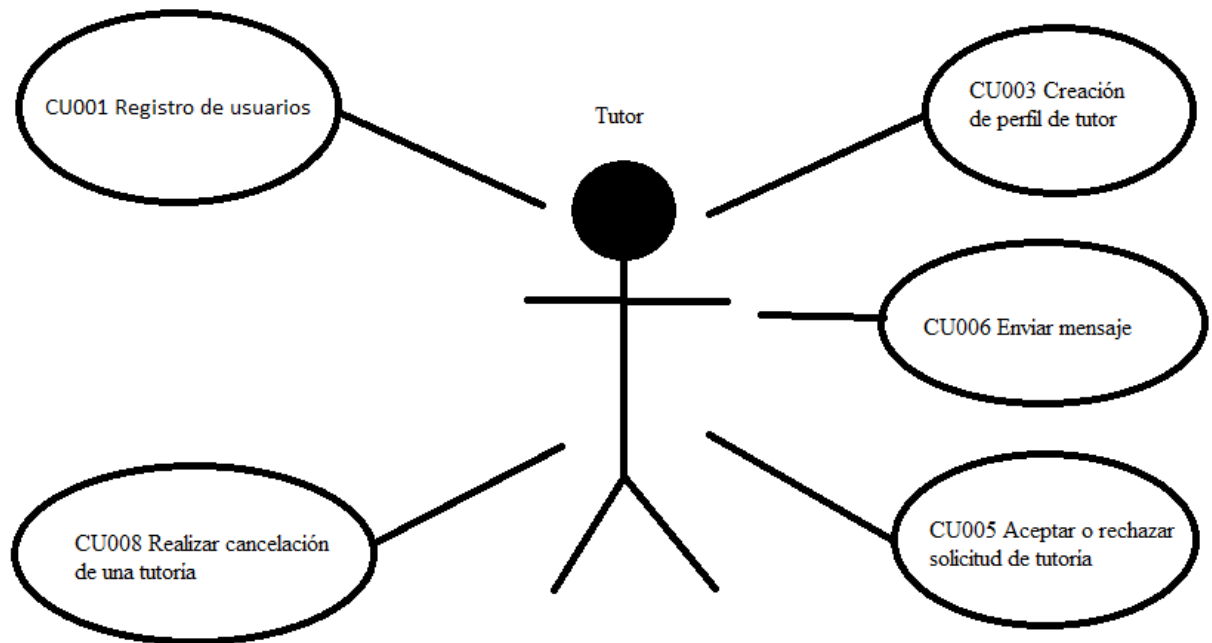
5.1.6 Diagramas de casos de uso

A continuación, se presentan los diagramas de casos de uso del sistema de relacionamiento para tutorías universitarias desde la perspectiva de los usuarios estudiantes y tutores.

Figura 7 Diagrama de casos de uso de estudiante



Fuente: Elaboración propia.

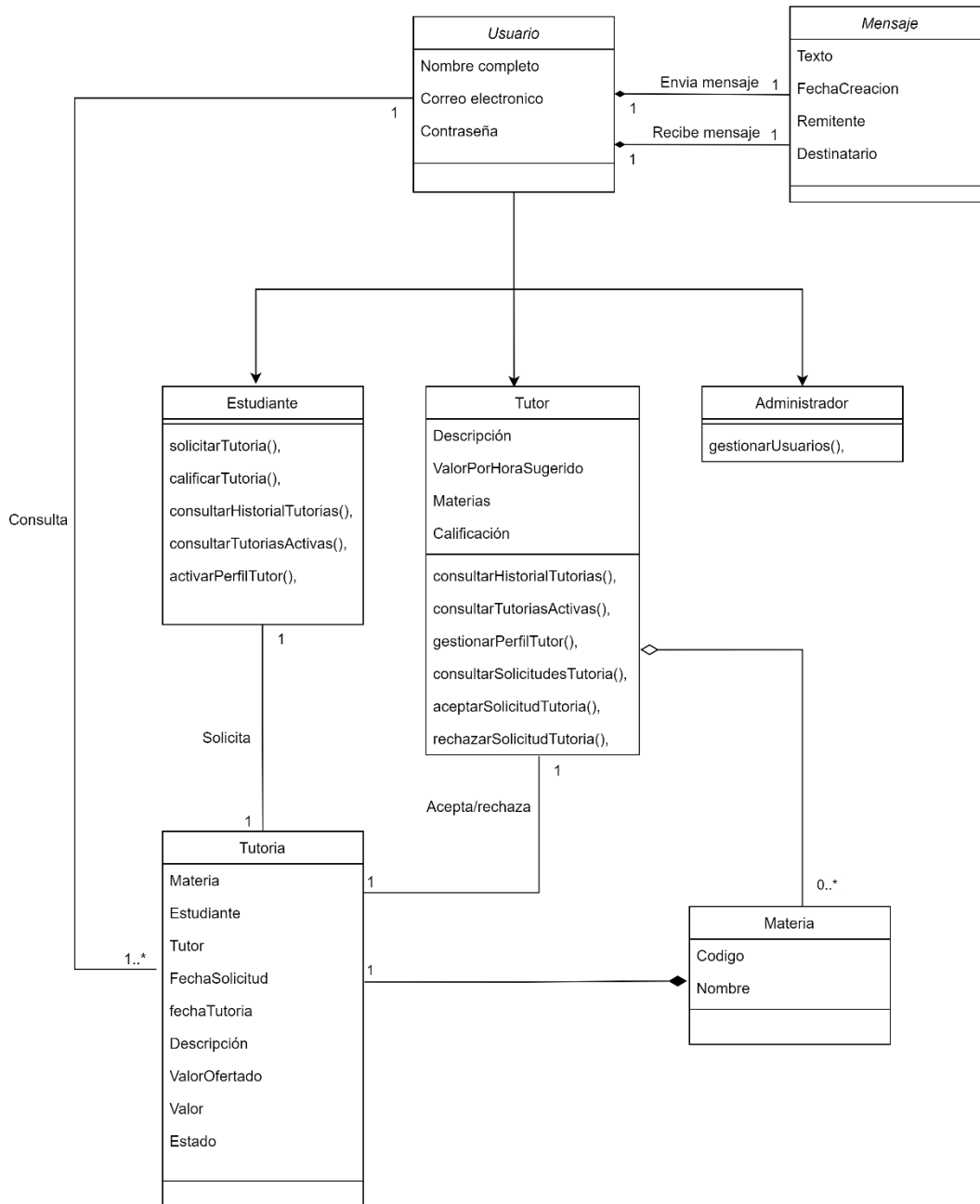
Figura 8 Diagrama de casos de uso tutor

Fuente: Elaboración propia.

5.1.7 Modelo de objetos

La siguiente imagen representa el diagrama de objetos de la plataforma de tutorías. Este diagrama visualiza las clases y las interacciones entre ellas, brindando una representación visual de la estructura y el flujo de datos en el sistema. A través de este diagrama, se puede comprender mejor cómo se organizan los componentes de la plataforma y cómo interactúan entre sí para proporcionar las funcionalidades necesarias a los usuarios.

Figura 9 Modelo de objetos del sistema



Fuente: Elaboración propia.

5.2 Diseño del sistema

El diseño del sistema se ha realizado con el objetivo de lograr una arquitectura escalable y fácilmente extensible. Se ha puesto especial atención en la definición clara de las responsabilidades y funcionalidades de cada componente, con el propósito de facilitar su mantenimiento y permitir su evolución a futuro.

5.2.1 Arquitectura del sistema.

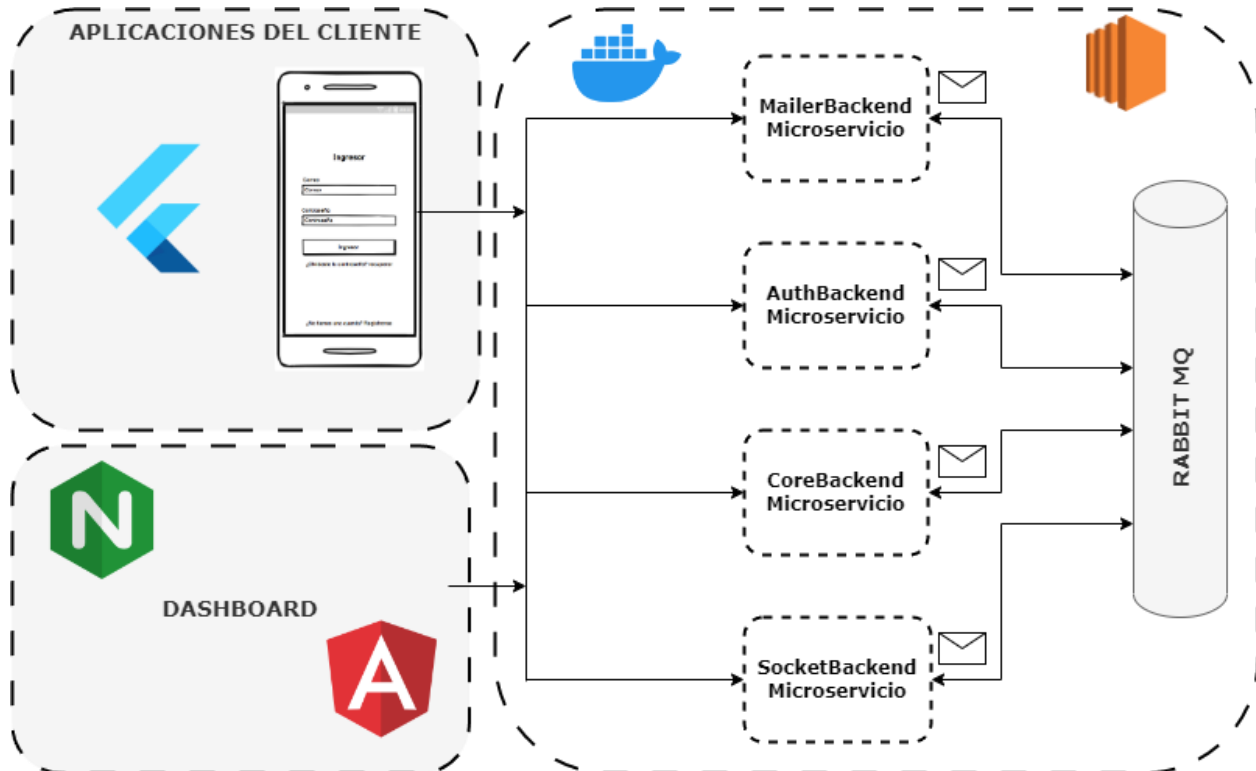
La arquitectura del sistema se basa en una estructura de microservicios, lo que permite una mayor modularidad y flexibilidad en el desarrollo y mantenimiento del sistema. Cada microservicio backend se ha implementado utilizando el framework NestJS, que proporciona una estructura escalable y orientada a eventos. La comunicación entre los microservicios se realiza a través de RabbitMQ, un sistema de mensajería que garantiza una comunicación eficiente y asíncrona entre los diferentes componentes del sistema.

En cuanto a la persistencia de datos, se ha tomado la decisión de emplear PostgreSQL como sistema de gestión de base de datos. La elección de PostgreSQL se basa en su confiabilidad y estabilidad, en su escalabilidad y en la garantía de la integridad de los datos.

Para el desarrollo de la aplicación móvil, se ha utilizado el framework Flutter. La elección de Flutter se basa en su naturaleza multiplataforma, lo que permite crear una única base de código que puede ser utilizada tanto en dispositivos Android como iOS. Esta capacidad de desarrollo multiplataforma ahorra tiempo y recursos al equipo de desarrollo.

A continuación, se presenta un diagrama de arquitectura del sistema:

Figura 10 Arquitectura del sistema



Fuente: Elaboración propia.

5.2.2 Modelado de procesos del sistema.

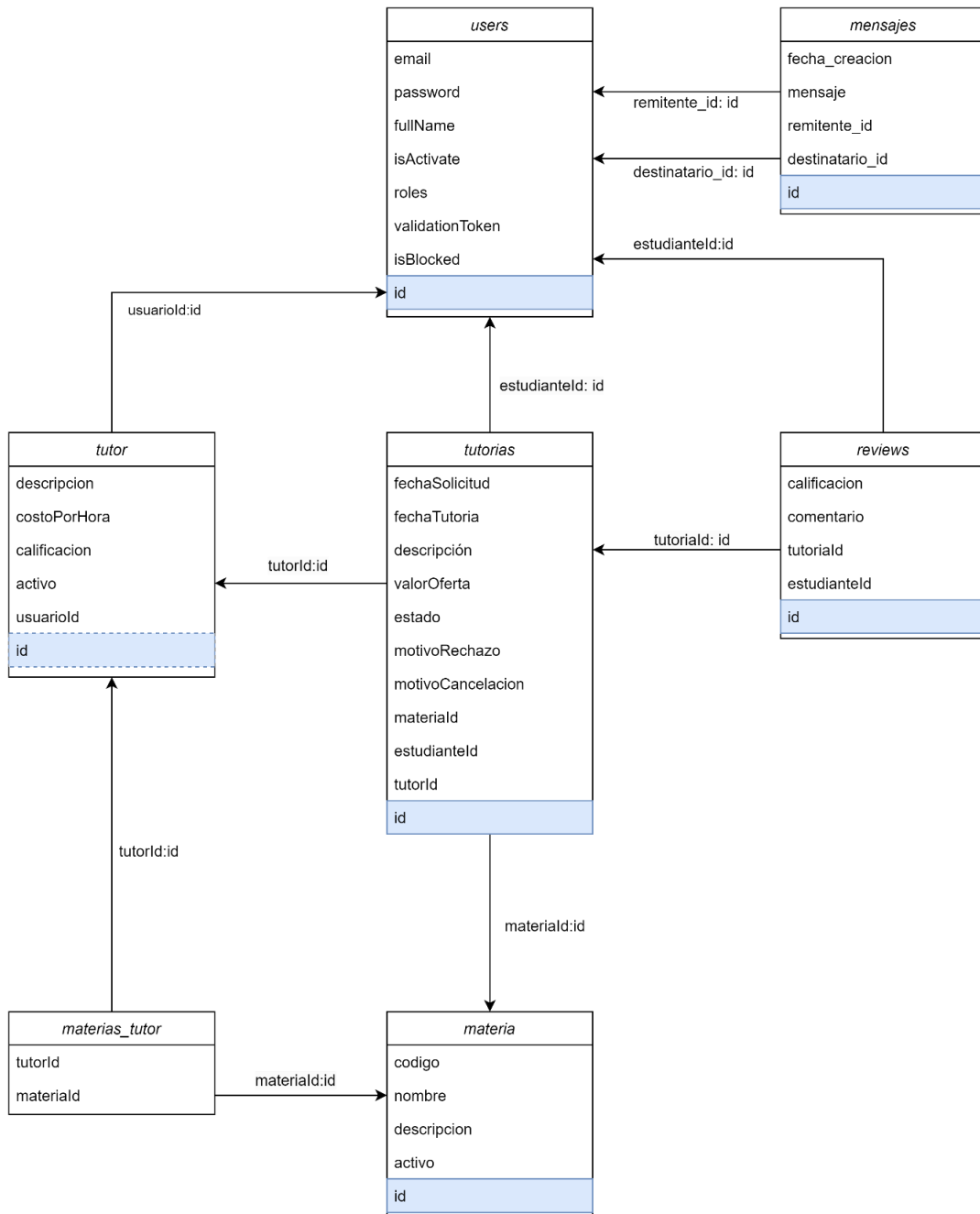
Durante el desarrollo de este proyecto, fue fundamental la creación de diversos diagramas para modelar los procesos y flujos de la plataforma. Estos diagramas permitieron visualizar de manera clara y concisa cómo interactúan los diferentes componentes y cómo se llevan a cabo las diferentes acciones dentro del sistema. En la sección de apéndice de este proyecto se encuentran disponibles dichos diagramas, los cuales brindan una perspectiva detallada de la estructura y funcionamiento de la plataforma.

5.2.3 Diseño de prototipos de interfaz de usuario

Se han desarrollado prototipos de interfaz de usuario para el sistema con el objetivo de proporcionar una representación visual de cómo podría verse y funcionar el sistema. Sin embargo, es importante tener en cuenta que los autores del proyecto no tienen conocimientos especializados en diseño de interfaces ni en experiencia de usuario, lo que implica que los diseños son provisionales y pueden mejorarse con la ayuda de un experto en la materia. Para crear los diseños, se utilizó la herramienta en línea Balsamiq Cloud. Se recomienda realizar una validación de la interfaz de usuario para asegurarse de que el sistema sea fácil de usar y accesible para los usuarios. Los diseños podrán ser consultados en la sección de apéndice de este proyecto.

5.2.4 Modelo de base de datos

Figura 11 Modelo de base de datos del sistema



Fuente: Elaboración propia.

5.3 Implementación

A continuación, se explicará a detalle cómo fue desarrollado el prototipo de sistema de tutorías, el código fuente podrá ser consultado en los siguientes repositorios:

- [Horus19/TUTUIS: Codigo fuente de sistema de tutorias para proyecto de grado \(github.com\)](https://github.com/Horus19/TUTUIS)
- [Horus19/chat_app \(github.com\)](https://github.com/Horus19/chat_app)

5.3.1 Autenticación de los usuarios.

El módulo de autenticación es uno de los componentes más importantes del sistema de relacionamiento para tutorías universitarias, ya que permite que los usuarios accedan al sistema y hagan uso de sus funcionalidades de forma segura. Este módulo es el encargado de gestionar la autenticación y autorización de los usuarios, lo que significa que garantiza que solo los usuarios autorizados puedan acceder a la información y recursos del sistema.

Para este módulo, fue necesario crear un microservicio dedicado exclusivamente a la autenticación en el sistema. En este microservicio se implementaron varias herramientas para garantizar la seguridad de la autenticación de los usuarios. En primer lugar, se utilizó Json Web Tokens (JWT) para permitir la autenticación y autorización de los usuarios. Además, se implementó el algoritmo bcrypt para el hash y la comparación de contraseñas de manera segura. También se utilizó Bearer Tokens para autenticar las solicitudes a las API y protegerlas contra accesos no autorizados.

Por último, se implementaron Custom Class Decorators utilizando AuthGuard del paquete nestjs/passport para verificar la autenticación y autorización de los usuarios en las rutas específicas del sistema. Por parte de la aplicación móvil, se usaron paquetes como http dart para realizar las peticiones a los microservicios, y flutter_secure_storage para almacenar el token de sesión y algunos otros valores importantes para manejar la sesión. Para la implementación del correo de bienvenida, se utilizó el microservicio de envío de correos que utiliza el paquete '@nestjs-modules/mailer' y el servicio de correo Amazon SES. Gracias a esto, se logró enviar un correo de bienvenida a los nuevos usuarios registrados en la plataforma.

5.3.2 Envío de correos

Durante el proceso de desarrollo del sistema, se identificó la necesidad de enviar correos electrónicos de forma automatizada a los usuarios registrados en la plataforma. Para solucionar esta necesidad, se optó por desarrollar un microservicio especializado en el envío de correos. Para el desarrollo del microservicio se utilizó el framework NestJS y el paquete NodeMailer con el protocolo SMTP.

Como proveedor de servicio de correo se tuvieron en cuenta a Mailgun y Amazon Simple Email Service (SES), decidiendo finalmente utilizar Amazon SES debido a su escalabilidad y facilidad de integración con otros servicios en la nube de Amazon. Además, se implementó la configuración de las credenciales y el host del servicio de correo como variables de entorno en la imagen Docker, permitiendo una facilidad para cambiar de proveedor.

Además de la creación del microservicio de envío de correos, también fue necesario desarrollar plantillas para los correos electrónicos. Estas plantillas están diseñadas con la extensión HBS y contienen variables que se reemplazan en tiempo de ejecución con información específica del usuario o del contenido del correo. En la sección de apéndices se muestra un ejemplo de esas plantillas.

5.3.3 Comunicación entre los microservicios

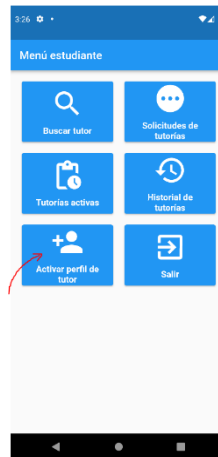
Durante el desarrollo del sistema también se presentó la necesidad de establecer una comunicación eficiente entre los diferentes microservicios que lo conforman. Por ejemplo, era necesario que el microservicio de autenticación pudiera dar la orden al microservicio de envío de correos con los datos del usuario para enviar el correo de bienvenida cuando se registraba en la plataforma. Para lograr esto, se decidió utilizar protocolo AMQP, implementado a través del servidor de mensajería RabbitMQ, una herramienta de mensajería que permite la comunicación asíncrona entre los distintos componentes del sistema.

En cada microservicio fue necesaria la creación de servicios que se encargaran de hacer la conexión con el servidor RabbitMQ y que pudieran enviar y suscribirse a las colas de mensajes, para esto, se utilizó la librería amqplib de Node.js. A continuación, se mostrará como se implementaron estos servicios.

5.3.4 Gestión de perfil de tutor

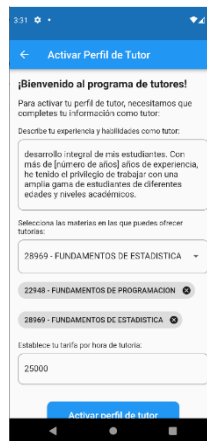
Por defecto, cuando un usuario se registra en la plataforma solo cuenta con el rol de estudiante activado, de esta forma cuando ingresa a la aplicación es dirigido directamente al menú de estudiante. Si un estudiante desea convertirse en tutor, se ha desarrollado un formulario para activar el perfil de tutor, el cual puede ser accedido a través de la opción “Activar perfil de tutor”.

Figura 12 Opción para activar perfil de tutor



Fuente: Elaboración propia.

Al pulsar sobre esta opción, será redirigido a una ventana donde se le solicitarán los datos básicos para la creación del perfil de tutor, tales como una descripción de su perfil, las materias de las cuales puede dictar las tutorías y un precio estimado por hora de tutoría.

Figura 13 Formulario de activación de perfil de tutor

3:31

← Activar Perfil de Tutor

¡Bienvenido al programa de tutores!

Para activar tu perfil de tutor, necesitamos que completes tu información como tutor:

Describe tu experiencia y habilidades como tutor:

desarrollo integral de mis estudiantes. Con más de [número de años] años de experiencia, he tenido el privilegio de trabajar con una amplia gama de estudiantes de diferentes edades y niveles académicos.

Selecciona las materias en las que puedes ofrecer tutorías:

28969 - FUNDAMENTOS DE ESTADISTICA

22948 - FUNDAMENTOS DE PROGRAMACION

28969 - FUNDAMENTOS DE ESTADISTICA

Establece tu tarifa por hora de tutoría:

25000

Activar perfil de tutor

Fuente: Elaboración propia.

Si el tutor desea editar su información como tutor, podrá hacerlo desde el menú de tutor y pulsando sobre la opción “gestionar perfil”. Desde esta opción, el usuario podrá cambiar su descripción, el precio sugerido o la lista de materias de las cuales puede dictar tutorías.

Figura 14 Formulario de gestión de perfil

3:47

← Gestionar perfil

Editar información del tutor:

Aquí podrás actualizar los datos de tu perfil de tutor. Por favor asegúrate de revisar cuidadosamente toda la información antes de guardar los cambios.

Descripción, experiencia y habilidades como tutor:

Soy un apasionado tutor con una sólida experiencia en la educación y un enfoque en el desarrollo integral de mis estudiantes. Con más de [número de años] años de experiencia, he tenido el privilegio de trabajar con una

Materias en las que puedes ofrecer tutorías:

Selecciona...

28969 - FUNDAMENTOS DE ESTADISTICA

22948 - FUNDAMENTOS DE PROGRAMACION

Tarifa sugerida por hora de tutoría

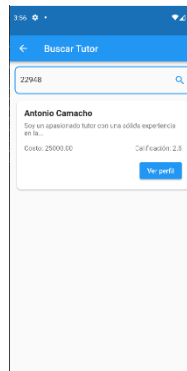
25000.00

Fuente: Elaboración propia.

5.3.5 Solicitud de tutoría

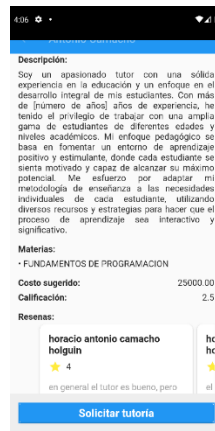
Para solicitar una tutoría, el usuario debe ingresar con el rol de estudiante y pulsar sobre la opción “buscar tutor”. Dentro de esta opción, el estudiante podrá buscar el tutor por su nombre, el nombre de la materia o por el código de la materia. Al ingresar un criterio de búsqueda, la aplicación le mostrará un listado de tutores que coinciden, presentando por cada tutor, su nombre, una breve introducción a su descripción, su valor sugerido y la calificación que presenta al momento.

Figura 15 Búsqueda del tutor



Fuente: Elaboración propia.

Al pulsar sobre “ver perfil”, el estudiante podrá ver la información detallada del tutor, como su descripción completa, la lista de materias de las cuales dicta tutorías, el precio que propone por hora de tutoría, la calificación que ha obtenido por sus tutorías y las reseñas que otros estudiantes han hecho sobre las tutorías dictadas por el tutor.

Figura 16 Detalles del tutor

Fuente: Elaboración propia.

Si el estudiante pulsa sobre el botón de “solicitar tutoría”, la aplicación redireccionará a una pantalla con un formulario básico para crear la solicitud de tutoría, en este formulario se le pide al estudiante que seleccione la materia de la que desea recibir la tutoría, que ingrese el valor que ofrece por la tutoría, seleccione la fecha y hora de cuándo será la tutoría y que proporcione una descripción de la tutoría, en donde podrá detallar el tema que necesita aprender, el lugar en donde necesita la tutoría y demás detalles que considere convenientes.

Figura 17 Formulario de solicitud de tutoría

4:17

Solicitud de Tutoría

Para solicitar una tutoría, por favor diligencie el siguiente formulario:

Seleccione la materia en la que desea recibir tutoría:

Materia

FUNDAMENTOS DE PROGRAMACION

Describa brevemente el tema sobre el que desea recibir tutoría:

Descripción

He tomado conocimiento de su destacada trayectoria como tutor/a en este campo y me gustaría aprovechar la oportunidad de recibir su orientación y apoyo en mi proceso de aprendizaje. Después de investigar y revisar

Ingrese el valor que ofrece por la tutoría:

Valor de la oferta

25000

Seleccione la fecha y hora de la tutoría:

Fecha y hora de la tutoría

18 de mayo de 2023 18:00

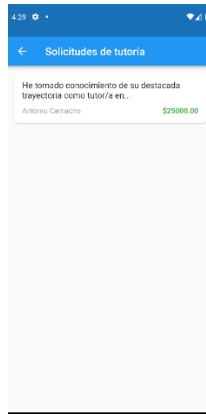
Enviar Solicitud

Fuente: Elaboración propia.

Si el estudiante desea cancelar una solicitud de tutoría, puede hacerlo en la lista de solicitudes. Desde el menú de estudiante puede pulsar sobre “Solicitudes de tutorías”, la aplicación redireccionará a una ventana donde se encuentran las solicitudes de tutorías que aún no han sido aceptadas o rechazadas.

Cada solicitud en la lista de solicitudes de tutorías muestra una introducción de la descripción de la solicitud, el valor y el tutor a quien se ha solicitado la tutoría.

Figura 18 Listado de solicitudes de tutoría



Fuente: Elaboración propia.

Al pulsar sobre la solicitud, la aplicación redirecciona a una pantalla donde el estudiante podrá ver los detalles de la solicitud realizada.

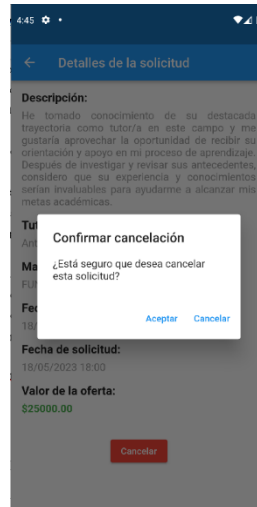
Figura 19 Detalles de solicitud



Fuente: Elaboración propia.

Si el estudiante pulsa sobre la opción “Cancelar”, aparecerá una ventana modal de confirmación para asegurar que el estudiante no cancele la solicitud por error.

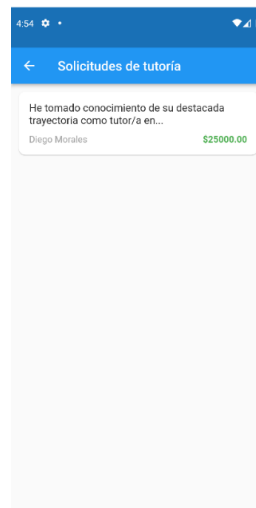
Figura 20 Confirmación de cancelación de tutoría



Fuente: Elaboración propia.

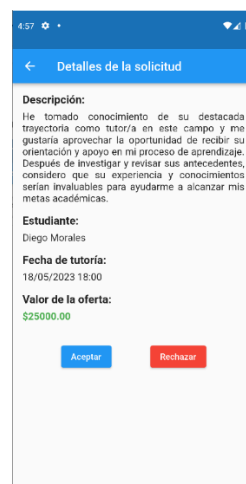
5.3.6 Gestión de solicitudes de tutoría

Desde el menú del tutor, al pulsar sobre la opción “Solicitudes de tutoría”, el usuario puede consultar las solicitudes de tutoría que se le han hecho. Desde esta opción, el usuario podrá ver el listado de solicitudes de tutoría que se le han realizado, presentando por cada solicitud, una breve introducción a la descripción de la tutoría, el nombre el estudiante que solicita y el valor que ofrece por la tutoría.

Figura 21 Solicitudes de tutoría desde perfil de tutor

Fuente: Elaboración propia.

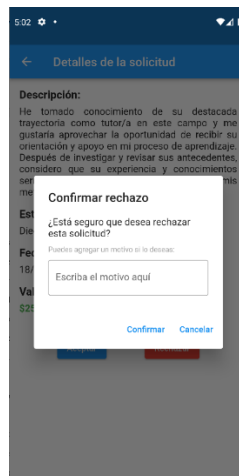
Al pulsar sobre una solicitud de tutoría de la lista, la aplicación le presentara una pantalla con los detalles de la solicitud y la opción de aceptar o rechazar la tutoría.

Figura 22 Detalle de solicitud desde perfil de tutor

Fuente: Elaboración propia.

Su el tutor pulsa sobre la opción aceptar, la aplicación mostrara un modal pidiendo que confirme la aceptación de la tutoría, si el tutor lo confirma, el estudiante es notificado y la tutoría pasa a un estado de aceptada. Si el tutor pulsa sobre rechazar, la aplicación mostrará un modal pidiendo que confirme el rechazo de la solicitud, pero además permitiéndole escribir un motivo del rechazo de la solicitud.

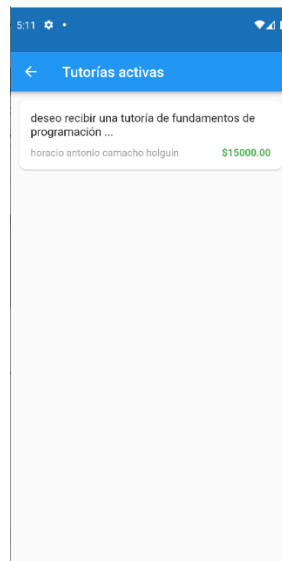
Figura 23 Confirmar rechazo de solicitud



Fuente: Elaboración propia.

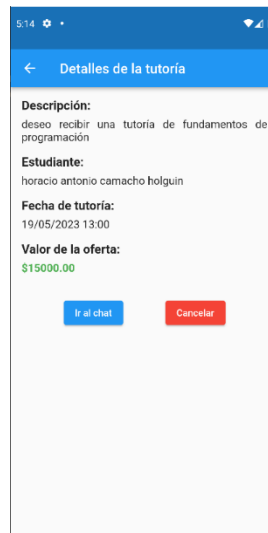
5.3.7 Tutorías activas

Como tutor o como estudiante, un usuario puede consultar las tutorías que tiene activas. Desde el menú de tutor, al pulsar en la opción “tutorías activas”, la aplicación mostrará un listado con las tutorías activas para el tutor.

Figura 24 tutorías activas de un tutor

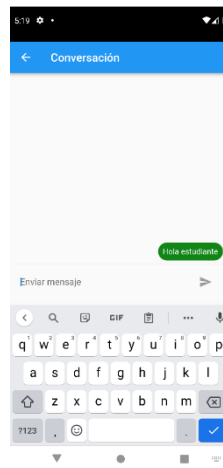
Fuente: Elaboración propia.

Al igual que en los demás listados, se presenta el nombre del estudiante, el valor y la descripción de la tutoría. Al pulsar sobre la tutoría, se navega a una pantalla donde se presentan los detalles de la tutoría y las opciones para ir al chat con el estudiante y si la fecha se encuentra dentro del rango aceptado en la lógica de negocio, también se presenta la opción de cancelar la tutoría.

Figura 25 Pantalla de tutoría activa

Fuente: Elaboración propia.

Si el usuario pulsa sobre “ir al chat”, la aplicación abrirá una pantalla en donde el tutor y el estudiante pueden comunicarse mientras la tutoría se encuentre activa.

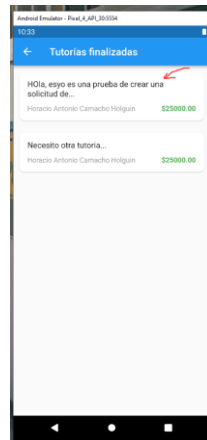
Figura 26 Pantalla de conversación entre estudiante y tutor

Fuente: Elaboración propia.

5.3.8 Calificación de tutorías

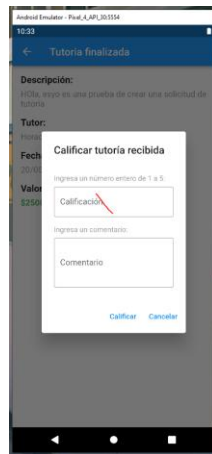
Como estudiante, una vez finalizada una tutoría, puede dejar su calificación para compartir su experiencia con los demás usuarios de la plataforma. Para calificar una tutoría, el estudiante puede pulsar sobre “Historial de tutorías” desde el menú de estudiantes, entonces la aplicación mostrara una vista con las tutorías finalizadas. En el listado de tutorías finalizadas el estudiante puede seleccionar la tutoría que desea calificar.

Figura 27 Listado de tutorías finalizadas



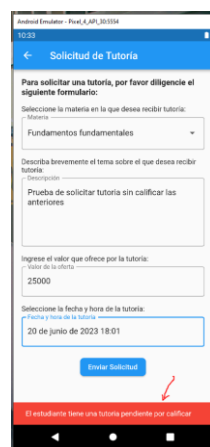
Fuente: Elaboración propia.

Al pulsar sobre una tutoría finalizada, la aplicación muestra una ventana con los detalles de la tutoría. Si se pulsa sobre “calificar”, se muestra un modal para agregar un comentario y una calificación.

Figura 28 Modal de calificación de tutoría

Fuente: Elaboración propia.

Al agregar la calificación desaparece la opción para calificar la tutoría y se actualiza la calificación general del tutor, además se podrá ver el comentario en el perfil del tutor. En caso de que el estudiante no califique las tutorías finalizadas, no podrá solicitar una tutoría más.

Figura 29 Solicitud de tutoría sin calificar las anteriores

Fuente: Elaboración propia.

5.4 Pruebas

5.4.1 Pruebas unitarias

Para la realización de pruebas unitarias en los microservicios desarrollados con nestjs se utilizó Jest, una herramienta de pruebas de javascript que proporciona una gran cantidad de funcionalidades para realizar pruebas unitarias, incluyendo la ejecución de pruebas en paralelo, generación de informes de cobertura de código y la integración con herramientas de integración continua. A continuación, se explicará cómo se utilizó Jest en el proyecto para garantizar la calidad y la fiabilidad de los microservicios, utilizando como ejemplo las pruebas hechas para el auth service del microservicio de autenticación.

Por defecto, al crear un servicio en NestJS, también se genera un archivo de pruebas con extensión spec.ts correspondiente al servicio creado. Sin embargo, en caso de que se haya eliminado este archivo accidentalmente, es posible crearlo de forma manual. En este archivo se definirán las diversas pruebas unitarias que se realizarán al servicio con el objetivo de verificar su correcto funcionamiento y detectar posibles errores.

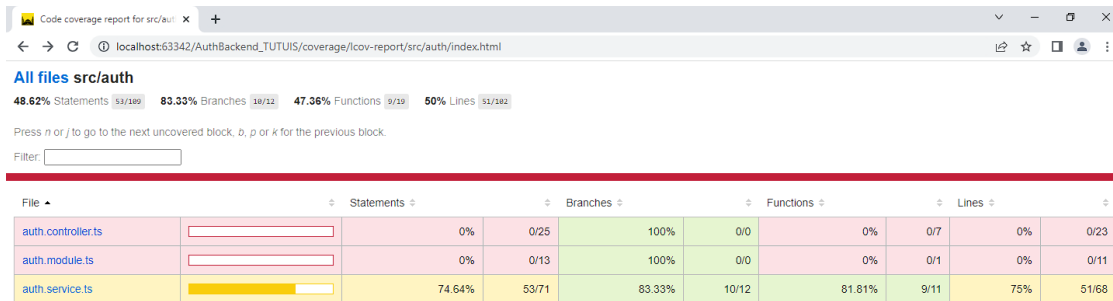
Cada prueba unitaria inicia con el describe que encapsula todas las pruebas hechas para el servicio, seguido de la declaración de las dependencias que necesita la prueba y del método beforeEach, en el cual se realiza la configuración inicial de las pruebas unitarias. El método beforeEach se ejecuta antes de cada prueba y se encarga de crear un módulo de pruebas que contiene los servicios y dependencias necesarios para las pruebas. La configuración se realiza a través del método createTestingModule, donde se definen los proveedores de los servicios y se

utiliza `jest.fn()` para simular el comportamiento de los métodos de los servicios. Luego, se utiliza el método `module.get()` para obtener una instancia de cada servicio y asignarla a una variable que se utilizará en las pruebas.

Una vez realizada la configuración inicial en el archivo de pruebas, se pueden crear los conjuntos de pruebas para cada método del servicio. Para agrupar las pruebas por método, se utilizó la función `describe`. De esta manera, se logra encapsular cada conjunto de pruebas para cada método del servicio y mantener una organización en las pruebas.

Para ejecutar las pruebas unitarias de un microservicio es necesario tener en cuenta que los demás microservicios deben estar desplegados, excepto el que se está probando, y que este último debe estar configurado para correr localmente. Una vez se cumplen estas condiciones, se pueden ejecutar las pruebas utilizando el comando `jest --coverage` en la raíz del microservicio. Este comando ejecutará las pruebas y generará un reporte de cobertura en la carpeta `coverage`. En dicha carpeta, se encontrará un archivo `index.html` que permite visualizar de manera gráfica qué partes del código han sido cubiertas por las pruebas unitarias.

Figura 30 Ejemplo de reporte de cobertura de las pruebas



Fuente: Elaboración propia.

Además de las estadísticas generales de las pruebas realizadas, el informe de cobertura también puede indicar visualmente las líneas de código que no han sido cubiertas por las pruebas, resaltándolas en color rojo para identificarlas fácilmente. De esta manera, se puede tener una idea clara de las áreas de código que necesitan más atención y pruebas para mejorar la calidad del software.

Figura 31 Cobertura de pruebas dentro del código

```

109  async changePassword(id: string, changePasswordDto: ChangePasswordDto) {
110  3x    const user = await this.userRepository.findOne({
111      1x      where: { id },
112      1x      select: ['id', 'email', 'password'],
113      1x    });
114
115  3x    if (!user) {
116  1x      throw new UnauthorizedException('Usuario no encontrado');
117    }
118
119  2x    if (!bcrypt.compareSync(changePasswordDto.password, user.password)) {
120  1x      throw new UnauthorizedException('Contraseña incorrecta');
121    }
122
123  1x    user.password = await bcrypt.hash(changePasswordDto.newPassword, 10);
124  1x    await this.userRepository.save(user);
125  1x    return {
126      1x      ...user,
127      1x      token: this.getJwtToken({ id: user.id }),
128    };
129  }
130
131  /// Metodo para activar el usuario
132  async activateUser(token: string) {
133  1x    if (!token) {
134  1x      throw new BadRequestException('Token no proporcionado');
135    }
136
137  1x    const user = await this.userRepository.findOne({
138      1x      where: { validationToken: token },
139    });
140
141  1x    const isValid = await this.verifyToken(token);
142  }

```

Fuente: Elaboración propia.

5.4.2 Pruebas de calidad en el código

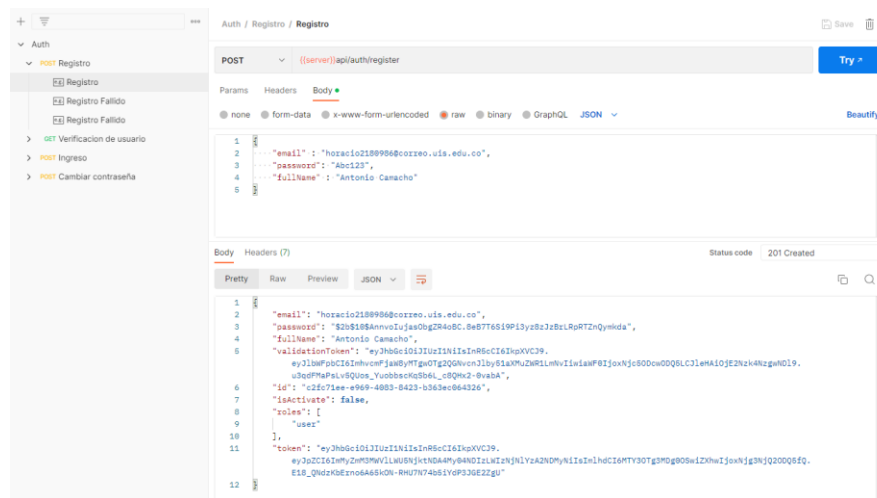
Para evaluar la calidad del código en el proyecto, se utilizaron dos herramientas principales: SonarLint y ESLint. SonarLint es una extensión de SonarQube que se integra con diversos IDEs y permite el análisis del código fuente en busca de vulnerabilidades, errores, malas prácticas y duplicaciones de código. Por otro lado, ESLint es una herramienta de análisis de código estático para JavaScript, que ayuda a mantener un código limpio y coherente.

5.5 Documentación

5.5.1 Documentación de API

La documentación de la API fue fundamental en este proyecto, permitiendo una conexión fluida entre los servicios backend y la aplicación que los consume. Además, es esencial para el futuro del proyecto, ya que facilita la comprensión de este a los nuevos desarrolladores que se integren al equipo.

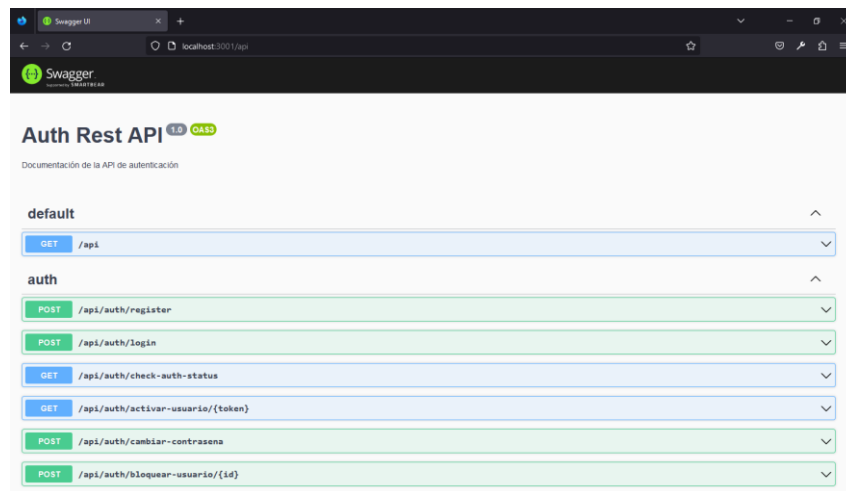
El primer método utilizado para documentar la API consistió en el uso de Postman. Se comenzó creando un workspace en la herramienta para guardar las colecciones correspondientes. Durante el desarrollo de los diferentes servicios, se realizaron pruebas utilizando Postman y se agregaron los endpoints correspondientes a la colección creada.

Figura 32 Ejemplo de colección en Postman

Fuente: Elaboración propia.

Además de la documentación de la API a través de Postman, se utilizó OpenAPI para documentar los microservicios. Para ello se hizo uso del paquete `@nestjs/swagger` que permite generar una documentación interactiva y dinámica de la API en la ruta `/api` de cada microservicio.

Cada microservicio tiene una ruta `/api` donde se encuentra la documentación generada por Swagger. Al ingresar a esta ruta en un navegador se mostrará una interfaz gráfica con los endpoints y sus descripciones correspondientes. Además, se pueden realizar pruebas de los endpoints desde esta interfaz, lo que facilita el desarrollo y la comprensión de cada microservicio.

Figura 33 Ejemplo de interfaz para consultar el api de cada microservicio

Fuente: Elaboración propia.

5.5.2 Documentación de código

Tanto el código de los microservicios como el código de la aplicación móvil cuentan con comentarios de tipo docstring, los cuales son una herramienta de documentación utilizada en la programación para facilitar el entendimiento del código. Estos comentarios, que se encuentran en la parte superior de clases, métodos o funciones, describen en detalle qué hace el código, qué parámetros recibe, qué devuelve y cómo se debe utilizar. De esta manera, los desarrolladores que trabajen con el código podrán entender fácilmente su funcionalidad y utilizarlo de manera correcta.

5.6 Despliegue

Para que la aplicación móvil pudiera acceder al backend de la plataforma, fue necesario desplegarlo. A continuación, se explicarán las configuraciones y pasos realizados para llevar a cabo el despliegue del sistema.

5.6.1 Creación de imágenes Docker

Para simplificar el proceso de creación y manejo de múltiples contenedores al mismo tiempo, se utilizó la herramienta Docker Compose. En general, los Dockerfiles para la creación de las imágenes de cada microservicio son muy similares, ya que en este proyecto se utilizó NestJS como framework de backend. Una de las diferencias notables entre los Dockerfiles es que, en el caso del servicio de envío de correos, fue necesario copiar los templates de los correos dentro de la imagen mediante el comando COPY.

La imagen de Docker del dashboard también es diferente a las demás creadas para este proyecto. Este servicio se basa en Angular, por lo que se utilizó una imagen base de Nginx para crear el Dockerfile correspondiente. En este archivo, se copió el código fuente del dashboard al contenedor de Docker y se realizó la compilación antes de servir los archivos estáticos mediante Nginx.

Para la creación de las imágenes y para correr el proyecto localmente se utilizó Docker Compose. Esta es una herramienta que permite definir y ejecutar múltiples contenedores Docker de forma simultánea. En el archivo de configuración de Docker Compose se especifica qué

servicios se deben crear, qué imágenes usar, qué puertos deben estar expuestos, entre otros detalles.

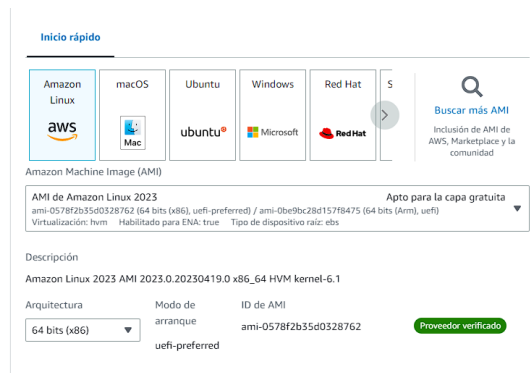
Una vez creadas las imágenes de los servicios, estas fueron subidas a docker hub para poder acceder fácilmente a ellas durante la configuración de despliegue. Finalmente, para levantar el proyecto se corre el siguiente comando en la raíz del proyecto:

```
docker-compose -f docker-compose.prod.yaml --env-file .env.prod up --build
```

5.6.2 Creación de instancia en el servicio EC2 de Amazon web services (AWS)

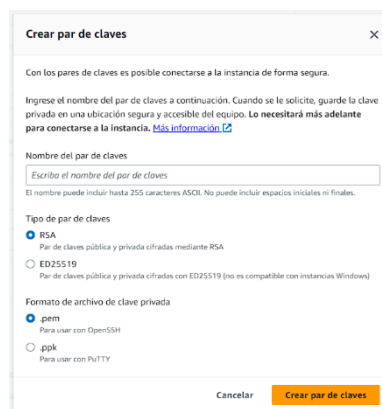
Para desplegar el sistema se decidió utilizar el servicio EC2 de Amazon, el cual ofrece una capa gratuita. A continuación, se explicará cómo se creó la instancia para desplegar el sistema.

El primer paso para crear la instancia de EC2 es seleccionar el tipo de instancia que se ajuste a las necesidades del proyecto. Para este caso, se decidió utilizar la versión de Amazon Linux 2023 como sistema operativo de la instancia. Este tipo de instancia se eligió debido a que estaba dentro de la capa gratuita ofrecida por Amazon y por ser suficiente para la demostración del prototipo. Sin embargo, se recomienda explorar otras opciones con mayor capacidad en el futuro.

Figura 34 Selección de tipo de instancia

Fuente: Elaboración propia.

El siguiente paso es configurar las claves pem que permitirán conectarse de manera segura a la instancia. Para ello, es necesario generar un nuevo par de claves pem en la consola de AWS y descargarlo en la máquina local. Luego, se debe modificar los permisos del archivo descargado para asegurarse de que sólo el propietario tenga acceso de lectura y escritura.

Figura 35 Creación de claves para conectarse a la instancia

Fuente: Elaboración propia.

El siguiente paso es muy importante para que los servicios creados puedan ser accedidos, a continuación, se muestra la creación de la configuración de red. En este caso, se creó una regla de tipo TCP personalizado con un tipo de origen de "anywhere" y con intervalos de puertos que van desde el 3000 hasta el 3008. Esto se hizo para permitir el acceso a los puertos expuestos por los servicios que se están ejecutando en la instancia EC2.

Figura 36 Creación de regla del grupo de seguridad

The screenshot shows the configuration for a security group rule. At the top, it displays 'Regla del grupo de seguridad 2 (TCP, 3000-3008, 0.0.0.0/0)' and an 'Eliminar' button. Below this, there are several fields: 'Tipo' (set to 'TCP personalizado'), 'Protocolo' (set to 'TCP'), 'Intervalo de puertos' (set to '3000-3008'), 'Tipo de origen' (set to 'Cualquier lugar'), 'Origen' (set to '0.0.0.0/0'), and 'Descripción - optional' (set to 'por ejemplo, SSH para Admin Desk'). Each field has an 'Información' link next to it. There is also a search bar for adding CIDR blocks and a list of prefixes.

Fuente: Elaboración propia.

El último paso en la creación de la instancia de EC2 es la configuración del almacenamiento. Por defecto, la instancia viene con 8 gigas de almacenamiento en su disco, sin embargo, para poder utilizar las imágenes Docker creadas y otras posibles necesidades del proyecto, es recomendable establecer una cantidad mayor de almacenamiento. Para este caso, se decidió aumentar el tamaño de almacenamiento a 16 gigas.

Figura 37 Configuración de almacenamiento instancia ec2

▼ Configurar almacenamiento Información Avanzado

1x 16 GiB gp3 Volumen raíz (Sin cifrar)

Los clientes que cumplan los requisitos de la capa gratuita pueden obtener hasta 30 GB de almacenamiento magnético o de uso general (SSD) de EBS

Agregar un nuevo volumen

0 x sistemas de archivos Editar

Fuente: Elaboración propia.

5.6.3 Configuración de instancia EC2

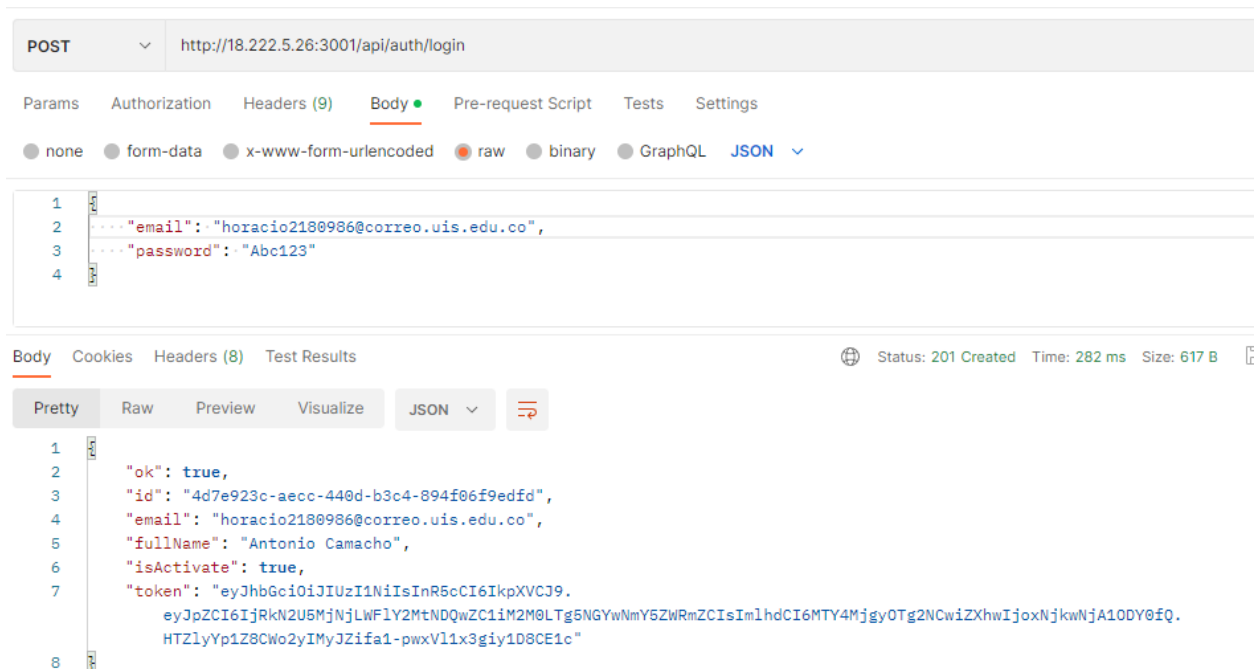
La configuración de la instancia EC2 requiere que se realice una conexión a ella, una opción es hacerlo mediante SSH. A continuación, se detallará el proceso para conectarse a la instancia de EC2 utilizando SSH. Es importante tener en cuenta que se debe tener acceso a la clave privada (.pem) utilizada en la creación de la instancia. El siguiente paso es instalar Docker en la instancia EC2. Una vez instalado Docker en la instancia EC2, el siguiente paso fue instalar Docker Compose. Docker Compose es una herramienta que permite definir y ejecutar aplicaciones Docker de múltiples contenedores. Para instalar Docker Compose, se ejecutaron los siguientes comandos:

5.6.4 Despliegue de proyecto en instancia EC2

Para desplegar el proyecto en la instancia de EC2 se utilizó un archivo de Docker Compose adicional, que es similar al utilizado para la creación de las imágenes, pero sin la instrucción "build". En este nuevo archivo se define cada uno de los servicios necesarios para el despliegue del sistema.

Para verificar que los servicios están en ejecución en la instancia de EC2, se puede utilizar la herramienta Postman. Para conocer la dirección IPv4 a la que se harán las solicitudes, se puede ir al resumen de la instancia en el panel de EC2.

Figura 38 Prueba de petición a servicio de auth en EC2



The screenshot displays a Postman interface for a POST request to `http://18.222.5.26:3001/api/auth/login`. The request body is a JSON object with the following content:

```
1 {
2   "email": "horacio2180986@correo.uis.edu.co",
3   "password": "Abc123"
4 }
```

The response body is a JSON object with the following content:

```
1 {
2   "ok": true,
3   "id": "4d7e923c-aecc-440d-b3c4-894f06f9edfd",
4   "email": "horacio2180986@correo.uis.edu.co",
5   "fullName": "Antonio Camacho",
6   "isActivate": true,
7   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjRkN2U5MjNjLWFLY2MtNDQwZC1iM2M0LTg5NGYwNmY5ZWRmZC1iIm1hdCI6MTY4Mjg2NCwiZXhwIjoxNjkwNjA1ODY0fQ.HTZ1yYp1Z8Cwo2yIMyJZifa1-pwxVl1x3giy1D8CE1c"
8 }
```

At the bottom right of the interface, the status is shown as "Status: 201 Created", with a response time of "Time: 282 ms" and a size of "Size: 617 B".

Fuente: Elaboración propia.

6. Conclusiones

Se logró el objetivo de diseñar e implementar el prototipo de un sistema software de relacionamiento para tutorías, siguiendo el diseño de arquitectura previamente analizado. La base de este desarrollo se estableció en los comentarios y recomendaciones proporcionados por los estudiantes en las encuestas realizadas, lo que permitió dar una solución que se ajusta a las necesidades descritas. El resultado es un prototipo versátil y fácil de usar, que facilita la comunicación entre tutores y estudiantes, mejorando la experiencia de las tutorías y proporcionando un valioso recurso educativo.

Después de finalizar el prototipo de sistema de agendamiento de tutorías, se ha comprobado la importancia y necesidad de contar con una plataforma que facilite la interacción entre tutores y estudiantes. La respuesta positiva y entusiasta de los usuarios valida la relevancia de esta herramienta y demuestra la existencia de una demanda real en el ámbito educativo. Estos resultados confirman que los estudiantes están dispuestos a utilizar este tipo de soluciones para mejorar su proceso de aprendizaje. Con base en estos hallazgos, se puede estar convencidos de que el sistema puede satisfacer eficazmente esta necesidad creciente.

Las recomendaciones recibidas durante la presentación del prototipo a un grupo de estudiantes han brindado una visión clara de las necesidades y expectativas de los estudiantes, lo cual permitirá mejorar y enriquecer la aplicación en futuras versiones.

7. Recomendaciones

Es importante tener en cuenta que los diseños de interfaz de usuario implementados en los prototipos son solo provisionales y deben ser validados por un especialista en experiencia de usuario. Se recomienda que se realicen pruebas de usabilidad y se realice una evaluación heurística para identificar posibles problemas en la interfaz de usuario. Además, se sugiere realizar iteraciones en el diseño basadas en los comentarios y sugerencias de los usuarios para mejorar la experiencia de usuario en el sistema.

También se debe tener en cuenta que la elección del tipo de instancia EC2 y del sistema operativo adecuado para el proyecto es crucial para su desempeño y escalabilidad. En este caso, se decidió utilizar una instancia de capa gratuita de Amazon Linux 2023 para reducir los costos de despliegue y demostrar el prototipo, pero se recomienda explorar opciones más robustas y escalables en proyectos más grandes y complejos. Es importante hacer una evaluación detallada de las necesidades del proyecto y de las opciones disponibles para seleccionar la instancia y sistema operativo adecuados.

Referencias Bibliográficas

- Aprender Gratis. (2022, September 18). *¿Qué es Coursera?* Aprender Gratis: Cursos, Guías, Tutoriales Y Manuales. <https://aprendergratis.es/que-es-coursera/>
- Atlassian. (n.d.). *What is Scrum?* [Video]. Atlassian. <https://www.atlassian.com/agile/scrum>
- B, G., & B, G. (2023). Qué es GitHub y cómo empezar a usarlo. *Tutoriales Hostinger*. <https://www.hostinger.es/tutoriales/que-es-github>
- Benedet, M. (2019). ¿Qué son los microservicios? Definición, características y retos. *#ADN CLOUD*. <https://blog.mdcloud.es/que-son-los-microservicios-definicion-caracteristicas-y-retos/>
- Bigelow, S. J., & Courtemanche, M. (2021). Docker. *ComputerWeekly.es*. <https://www.computerweekly.com/es/definicion/Docker>
- Chacón, J. L. (2022). TypeScript: qué es, diferencias con JavaScript y por qué aprenderlo. *Profile Software Services*. <https://profile.es/blog/que-es-typescript-vs-javascript/>
- Clean Code: código limpio, ¿qué es?* (n.d.). <https://www.hostgator.mx/blog/clean-code-codigo-limpio/>
- Documentation | NestJS - A progressive Node.js framework*. (n.d.). Documentation | NestJS - a Progressive Node.js Framework. <https://docs.nestjs.com/openapi/introduction>
- Epitech Spain. (2023). NestJS: ¿Qué es y por qué es interesante este framework Node.js? *Epitech Spain*. <https://www.epitech-it.es/nestjs-que-es/>
- ¿Es Posible Aprender Inglés con Preply? + Opiniones | Papora*. (n.d.). <https://www.papora.com/es/aprender-ingles/preply/>
- Getting Started · Jest*. (2023, March 6). <https://jestjs.io/docs/getting-started>
- Gigworker. (2020, December 2). Tutoría Wyzant: ¿Es una buena manera de obtener ingresos adicionales? | Gigonway. *Gigonway*. <https://gigonway.com/tutoria-wyzant/>

Hurtado, J. S. (2023, February 15). *Metodología Scrum: qué es y cómo utilizarla para acometer proyectos*. Thinking for Innovation. <https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/>

Introducción a Android Studio. (n.d.). *Android Developers*.
<https://developer.android.com/studio/intro?hl=es-419>

JavaScript With Syntax For Types. (n.d.). <https://www.typescriptlang.org/>

Kinsta. (2022, July 25). *¿Qué es Nest.js? Un Vistazo al Framework Ligero de JavaScript*. Kinsta®. <https://kinsta.com/es/base-de-conocimiento/nestjs/>

Linares, J. C. (2020). 1.- Qué es coursera? 2.- Cómo funciona? 3.-Opiniones 4.- Mejores Leer más. *Bit4learn*. <https://bit4learn.com/es/lms/coursera/>

Mendoza, F., & Mendoza, F. (2023). *¿Qué tan bueno es Coursera? AulaPro*.
<https://aulapro.co/revisiones/que-tan-bueno-es-coursera/>

Messaging that just works — RabbitMQ. (n.d.). <https://www.rabbitmq.com/>

NestJS - A progressive Node.js framework. (n.d.). NestJS - a Progressive Node.js Framework.
<https://nestjs.com/>

PostgreSQL. (n.d.). OVHcloud. <https://www.ovhcloud.com/es/lp/postgresql-definition/>

Presta, M. (2022). Flutter vs Dart: ¿cuál es mejor? *Back4App Blog*.
<https://blog.back4app.com/es/flutter-vs-dart-cual-es-mejor/>

QS World University Rankings 2023: Top Global Universities. (n.d.). Top Universities.
<https://www.topuniversities.com/university-rankings/world-university-rankings/2023>

¿Qué es Flutter? - Explicación de la aplicación Flutter - AWS. (n.d.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/flutter/>

¿Qué es la computación en la nube? | Google Cloud | Google Cloud. (n.d.). Google Cloud.
<https://cloud.google.com/learn/what-is-cloud-computing?hl=es-419>

Quiénes somos | Preply. (n.d.). <https://preply.com/es/quienes-somos>

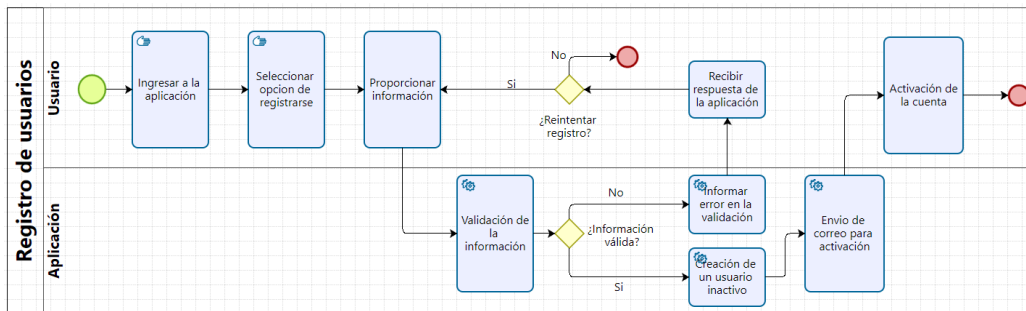
- Simões, C. (2021). ¿Qué es TypeScript, y por qué utilizarlo? *Blog ITDO - Agencia De Desarrollo Web, APPs Y Marketing En Barcelona*. <https://www.itdo.com/blog/que-es-typescript-y-por-que-utilizarlo/>
- Software Reservo. (2021). 3 ventajas de contar con sistema de agendamiento de citas. *reservo.cl*. <https://reservo.cl/blog/3-ventajas-de-contar-con-sistema-de-agendamiento-de-citas/>
- Software Reservo. (2022). ¿Por qué un sistema de agendamiento ayuda a tu empresa? *reservo.cl*. <https://reservo.cl/blog/por-que-un-sistema-de-agendamiento-ayuda-a-tu-empresa/>
- TablePlus. (2021, August 24). Capterra. <https://www.capterra.es/software/170642/tableplus>
- Tentulogo. (2022). Preply, conectando al mundo a través de la enseñanza. *Tentulogo*. <https://tentulogo.com/preply-conectando-al-mundo-a-traves-de-la-ensenanza/>
- Torres, A. (2021, December 15). Qué es SonarLint y cómo mejora la calidad de tu código. *excentia*. <https://www.excentia.es/que-es-sonarlint-y-como-mejora-calidad-codigo>
- Trustpilot. (2023, April 27). *Wyzant Opiniones*. <https://es.trustpilot.com/review/wyzant.com?languages=all>
- What is Amazon EC2? - Amazon Elastic Compute Cloud*. (n.d.). <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>

Apéndice

Apéndice A Modelos de procesos del sistema.

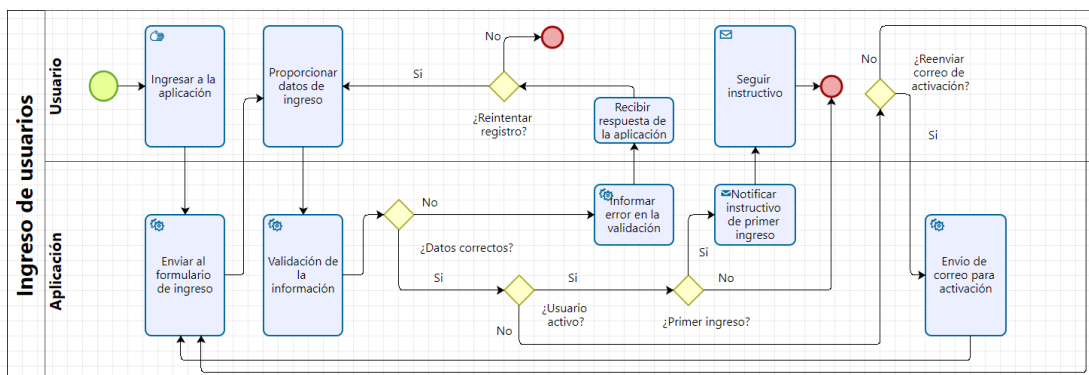
Se presentan los modelos de procesos de sistemas utilizados durante el desarrollo del proyecto. Estos modelos son representaciones gráficas que describen los diferentes procesos y flujos de trabajo de la plataforma de agendamiento de tutorías.

Figura 39 Modelado de registro de usuarios



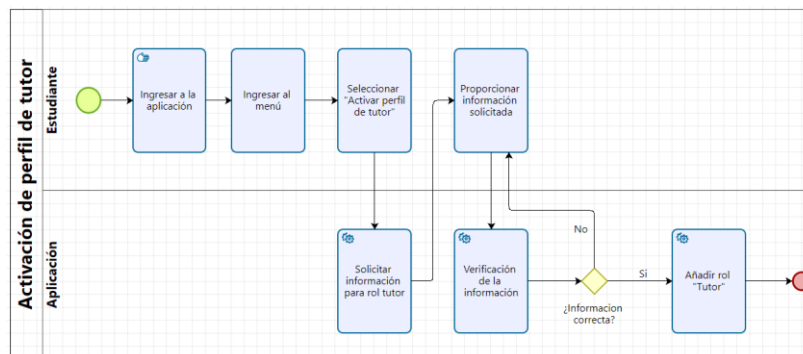
Fuente: Elaboración propia.

Figura 40 Modelado de ingreso de usuarios



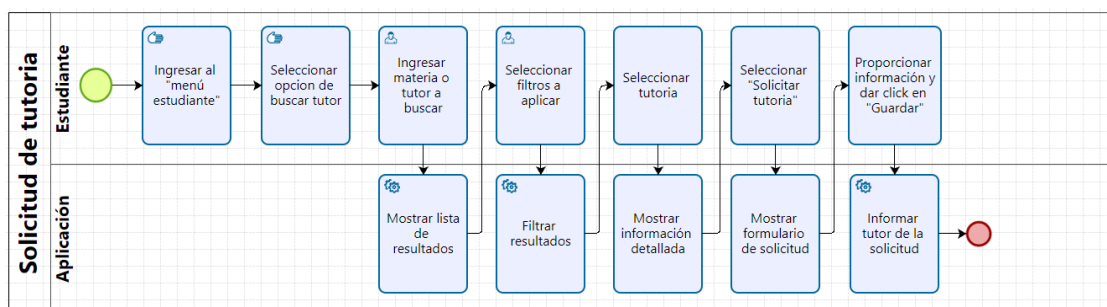
Fuente: Elaboración propia.

Figura 41 Modelado de activación de perfil de tutor



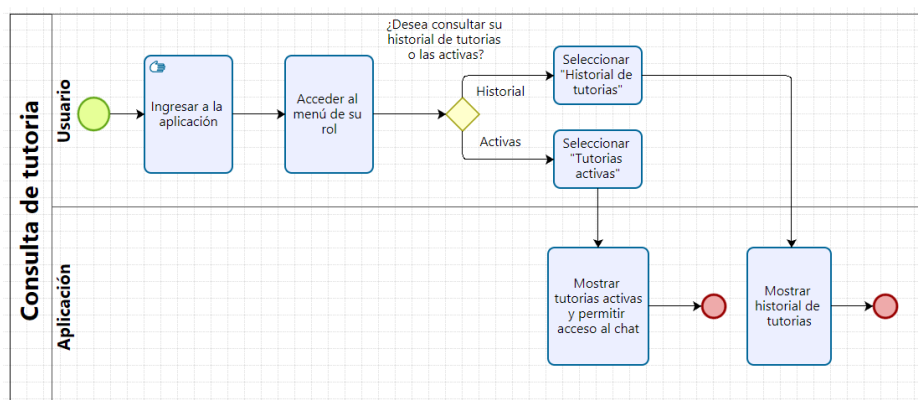
Fuente: Elaboración propia.

Figura 42 Modelado de solicitud de tutoría



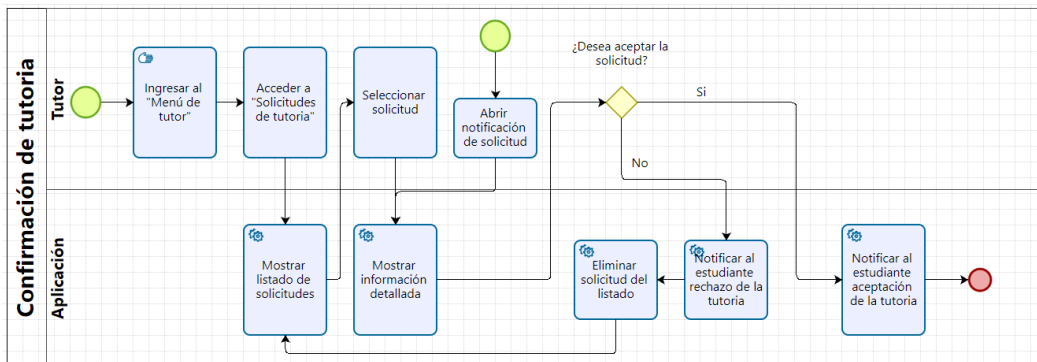
Fuente: Elaboración propia.

Figura 43 Modelado de consulta de tutorías



Fuente: Elaboración propia.

Figura 44 Modelado de confirmación de tutoría



Fuente: Elaboración propia.

Apéndice B Prototipos de interfaz

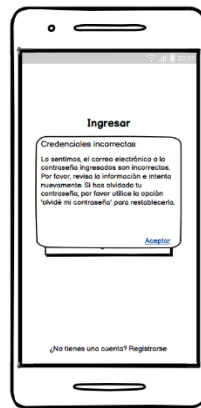
Se presentan los prototipos de interfaz desarrollados para la plataforma de agendamiento de tutorías. Estos prototipos son representaciones de las diferentes pantallas y elementos de la aplicación, que permiten tener una idea clara de cómo se verá y funcionará la interfaz final.

Figura 45 Prototipo para pantalla de ingreso



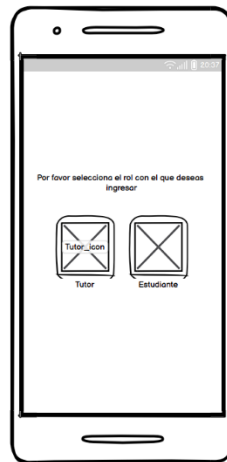
Fuente: Elaboración propia.

Figura 46 Prototipo para pantalla de ingreso con mensaje de error



Fuente: Elaboración propia.

Figura 47 Prototipo para pantalla de selección de rol para ingreso



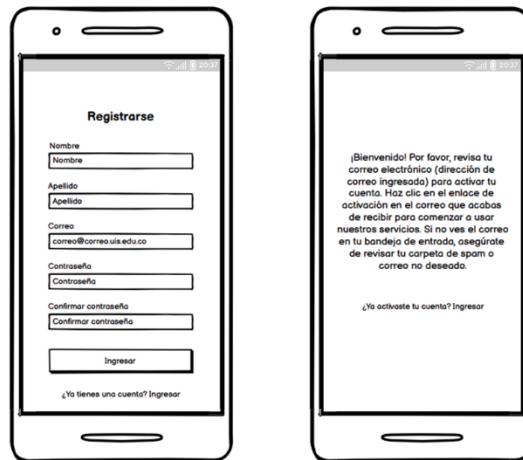
Fuente: Elaboración propia.

Figura 48 Prototipo para pantalla de primer ingreso de un usuario



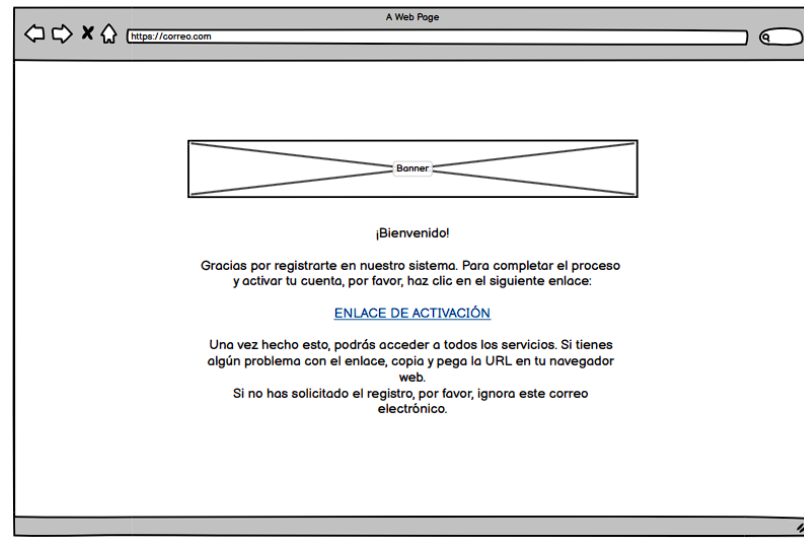
Fuente: Elaboración propia.

Figura 49 Prototipo para pantalla de mensaje de registro



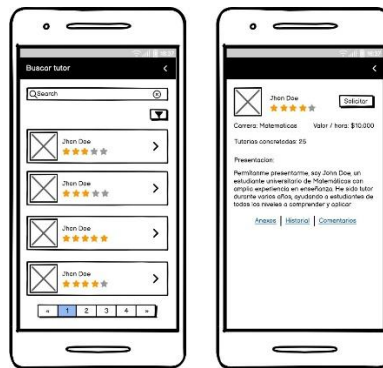
Fuente: Elaboración propia.

Figura 50 Prototipo de correo para activación de cuenta



Fuente: Elaboración propia.

Figura 51 Prototipo para consulta de tutor



Fuente: Elaboración propia.

Apéndice C Encuestas de usuario

Se presentan las encuestas realizadas para obtener las opiniones de los estudiantes sobre los requerimientos de la aplicación de tutorías. Estas encuestas han sido fundamentales para el éxito del proyecto, proporcionando datos cuantitativos y comentarios cualitativos que han

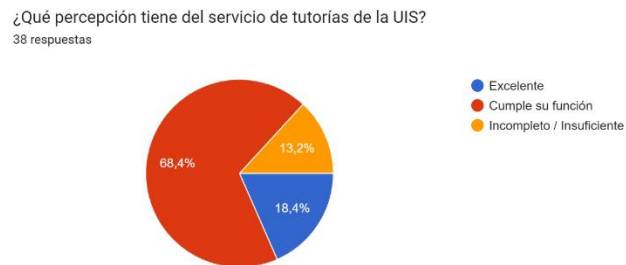
permitido ajustar los requerimientos y adaptar la solución a las necesidades de los estudiantes, garantizando una experiencia satisfactoria.

Figura 52 Respuestas primera pregunta, primera encuesta



Fuente: Elaboración propia.

Figura 53 Segunda pregunta, primera encuesta



Fuente: Elaboración propia.

Figura 54 Tercera pregunta, primera encuesta



Fuente: Elaboración propia.

Figura 55 Cuarta pregunta, primera encuesta



Fuente: Elaboración propia.

Figura 56 Quinta pregunta, primera encuesta



Fuente: Elaboración propia.

Figura 57 Sexta pregunta, primera encuesta

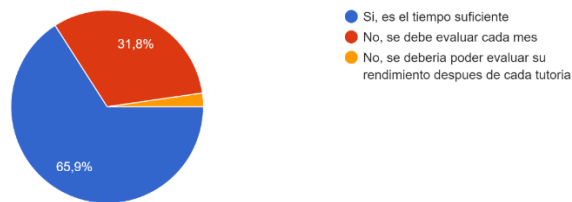
¿El sistema utilizado en las tutorías le parece el apropiado?
44 respuestas



Fuente: Elaboración propia.

Figura 58 Séptima pregunta, primera encuesta

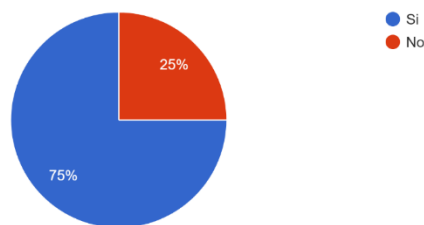
¿Considera correcto que se evalúe a los tutores al finalizar el semestre?
44 respuestas



Fuente: Elaboración propia.

Figura 59 Octava pregunta, primera encuesta

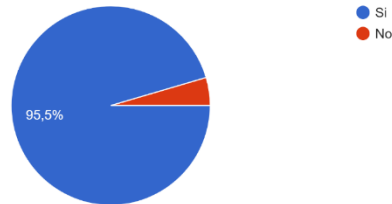
¿Los estudiantes deben poder elegir quien quieren que les dicte la tutoría?
44 respuestas



Fuente: Elaboración propia.

Figura 60 Novena pregunta, primera encuesta

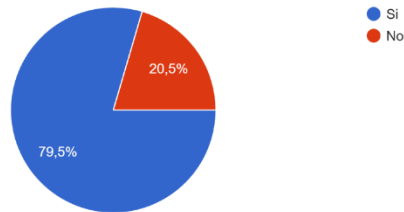
¿Tener un sistema de evaluación de tutores le daría más confianza a la hora de elegir a alguien para que le dicte la tutoría?
44 respuestas



Fuente: Elaboración propia.

Figura 61 Decima pregunta, primera encuesta

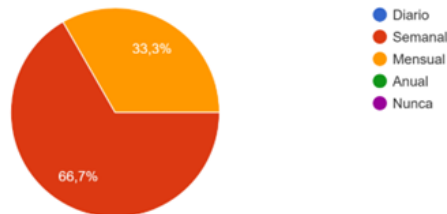
¿Los tutores deberían proporcionar su información (semestre, carrera, promedio en la asignatura que va a dictar, etc.) para que con base en ella los estudiantes puedan elegir quien les dicte la tutoría?
44 respuestas



Fuente: Elaboración propia.

Figura 62 Primera pregunta, segunda encuesta

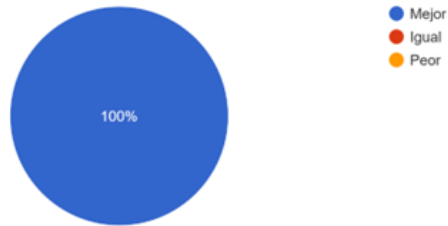
¿Qué tan seguido utilizaría la aplicación?
44 respuestas



Fuente: Elaboración propia.

Figura 63 Segunda pregunta, segunda encuesta

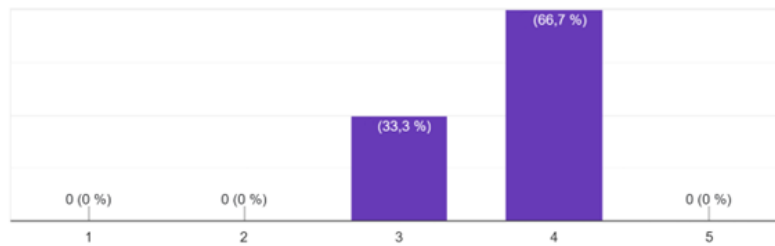
A comparación del sistema proporcionado por el SEA la aplicación provee de un servicio
44 respuestas



Fuente: Elaboración propia.

Figura 64 Tercera pregunta, segunda encuesta

¿Qué tan probable es que recomiende esta aplicación a un amigo?
44 respuestas



Fuente: Elaboración propia.