Desarrollo de una aplicación móvil como apoyo académico en ciencias básicas para el área de permanencia estudiantil de la universitaria Uniagustiniana

Manuel Alejandro Lozano Hernández

Cristóbal Fernando Pérez Chavarro

Universitaria Agustiniana - Uniagustiniana

Facultad de Ingeniería

Tecnología en Desarrollo de Software

Bogotá

2024

Desarrollo de una aplicación móvil como apoyo académico en ciencias básicas para el área de permanencia estudiantil de la universitaria Uniagustiniana

Manuel Alejandro Lozano Hernández

Cristóbal Fernando Pérez Chavarro

Tutor

Mauricio Alonso Villalba

Trabajo de grado para optar título de Tecnólogo en Desarrollo de Software

Universitaria Agustiniana - Uniagustiniana

Facultad de Ingeniería

Tecnología en Desarrollo de Software

Bogotá

2024

Tabla de contenido

Resumen (Abstract) 11

Abstract (resumen) 12

Introducción 13

1. Datos informativos del proyecto 15

1.1 Titulo del proyecto 15

1.2 Síntesis 15

2 Planificación del proyecto 16

2.1 Planteamiento del problema 16

2.2 Objetivos del proyecto 17

2.2.1 Objetivo general 17

2.2.2 Objetivos Específicos 17

2.3 Alcance de la solución 18

2.4 Metodología de desarrollo de software 18

2.4.1 Fase de exploración 19

2.4.2 Fase de inicialización 19

2.4.3 Fase de producción 20

2.4.4 Fase de estabilización 20

2.4.5 Fase de prueba 21

3 Especificación de requisitos de software (IEEE830) 22

3.1 Funcionalidad del producto 22
3.1.1 Registro y Gestión de Usuarios 22
3.1.2 Agendamiento de Tutorías 22
3.1.3 Gestión de Tutorías 23
3.1.4 Comunicación 23
3.1.5 Base de Datos 23
3.1.6 Interfaz de Usuario 24
3.2 Características de los Usuarios 24
3.3 Requisitos funcionales 25
3.4 Diagrama de casos de uso 26
3.5 Especificaciones de los casos de uso 27
3.6 Restricciones y Atributos de Calidad 32
4 Diseño del software (ISO-12207-1) 36 33
4.1 Diseño detallado del software 33
4.1.1 Diagrama de clases 33
4.1.2 Diagrama de componentes 34
4.1.3 Diagrama de actividades 35
4.1.4 Diagrama de despliegue 37
4.2 Diseño de la Interfaz 38

4.2.1 Interfaz de Usuario (Hardware, Software y Comunicaciones) 38

4.2.2 Interfaces de Entrada (Formularios, Log in) 49

4.2.3 Interfaces de Salida (Reportes, Consultas, Impresiones) 59

4.3 Diseño del modelo de datos / Persistencia (Modelo E-R) 64

4.4 Diseño de la Arquitectura de software (Modelo C4,nivel 1, nivel 2 y 3) 65

4.5 Diagrama de contenedores 66

4.6 Diagrama de componentes 67

5 Implementación 68

5.1 Herramientas utilizadas en el desarrollo del proyecto 68

5.2 Requisitos del hardware 68

6 Pruebas del software 70

6.1 Inspección de software (Validación y Verificación) (RQF-RQNF) 70

6.2 Pruebas de Usabilidad – Resultados 78

6.3 Heurísticas 83

7 Conclusiones y recomendaciones 94

8 Referencias 95

9 Anexos 96

9.1 Anexo A Manual de usuario 96

Lista de Tablas

Tabla 1 *Requisitos funcionales* 25

Tabla 2 *Tabla de registro* 27

Tabla 3 *Tabla de inicio de sesión* 28

Tabla 4 *Tabla de solicitud* 29

Tabla 5 *Tabla del chat* 30

Tabla 6 *Tabla de tutorías* 31

Tabla 7 *Tabla de Atributos de calidad.* 32

Lista de Figuras

Figura 1 *Diagrama de casos de uso* 26

Figura 2 *Diagrama de clases* 33

Figura 3 *Diagrama de componentes* 34

Figura 4 *Diagrama de actividades* 35

Figura 5 *Diagrama de actividades* 36

Figura 6 *Diagrama de Despliegue* 37

Figura 7 *Interfaz de Usuario* 38

Figura 8 *Interfaz de Usuario* 39

Figura 9 *Interfaz de Usuario* 40

Figura 10 *Interfaz de Usuario* 41

Figura 11 *Interfaz de Usuario* 42

Figura 12 *Interfaz de Usuario* 43

Figura 13 *Interfaz de Usuario* 44

Figura 14 *Interfaz de Usuario* 45

Figura 15 *Interfaz de Usuario* 46

Figura 16 *Interfaz de Usuario* 47

Figura 17 *Interfaz de Usuario* 48

Figura 18 *Interfaz de Entrada* 49

Figura 19 *Interfaz de Entrada* 50

Figura 20 *Interfaz de Entrada* 51

Figura 21 *Interfaz de Entrada* 52

Figura 22 *Interfaz de Entrada* 53

Figura 23 *Interfaz de Entrada* 54

Figura 24 *Interfaz de Entrada* 55

Figura 25 *Interfaz de Entrada* 56

Figura 26 *Interfaz de Entrada* 57

Figura 27 *Interfaz de Entrada* 58

Figura 28 *Interfaz de Salida* 59

Figura 29 *Interfaz de Salida* 60

Figura 30 *Interfaz de Salida* 61

Figura 31 *Interfaz de Salida* 62

Figura 32 *Interfaz de Salida* 63

Figura 33 *Diagrama de la Base de datos* 64

Figura 34 *Diagrama c4 Nivel 1* 65

Figura 35 *Diagrama c4 Nivel 2* 66

Figura 36 *Diagrama c4 Nivel 3* 67

Figura 37 *Inspección de Software* 70

Figura 38 *Inspección de Software* 70

Figura 39 *Inspección de Software* 71

Figura 40 *Inspección de Software* 71

Figura 41 *Inspección de Software* 71

Figura 42 *Inspección de Software* 72

Figura 43 *Inspección de Software* 72

Figura 44 *Inspección de Software* 72

Figura 45 *Inspección de Software* 73

Figura 46 *Inspección de Software* 73

Figura 47 *Inspección de Software* 73

Figura 48 *Inspección de Software* 74

Figura 49 *Inspección de Software* 74

Figura 50 *Inspección de Software* 74

Figura 51 *Inspección de Software* 75

Figura 52 *Inspección de Software* 75

Figura 53 *Inspección de Software* 75

Figura 54 *Inspección de Software* 76

Figura 55 *Inspección de Software* 76

Figura 56 *Inspección de Software* 76

Figura 57 *Inspección de Software* 77

Figura 58 *Inspección de Software* 77

Figura 59 *Inspección de Software* 77

Figura 60 *Pruebas de Usabilidad* 78

Figura 61 *Pruebas de Usabilidad* 79

Figura 62 *Pruebas de Usabilidad* 79

Figura 63 *Pruebas de Usabilidad* 79

Figura 64 *Pruebas de Usabilidad* 80

Figura 65 *Pruebas de Usabilidad* 80

Figura 66 *Pruebas de Usabilidad* 80

Figura 67 *Pruebas de Usabilidad* 81

Figura 68 *Pruebas de Usabilidad* 81

Figura 69 *Pruebas de Usabilidad* 81

Figura 70 *Pruebas de Usabilidad* 82

Figura 71 *Pruebas Heurísticas* 83

Figura 72 *Pruebas Heurísticas* 83

Figura 73 *Pruebas Heurísticas* 84

Figura 74 *Pruebas Heurísticas* 84

Figura 75 *Pruebas Heurísticas* 85

Figura 76 *Pruebas Heurísticas* 85

Figura 77 *Pruebas Heurísticas* 86

Figura 78 *Pruebas Heurísticas* 86

Figura 79 *Pruebas Heurísticas* 87

Figura 80 *Pruebas Heurísticas* 87

Figura 81 *Pruebas Heurísticas* 88

Figura 82 *Pruebas Heurísticas* 88

Figura 83 *Pruebas Heurísticas* 89

Figura 84 *Pruebas Heurísticas* 89

Figura 85 *Pruebas Heurísticas* 90

Figura 86 *Pruebas Heurísticas* 90

Figura 87 *Pruebas Heurísticas* 91

Figura 88 *Pruebas Heurísticas* 91

Figura 89 *Pruebas Heurísticas* 92

Figura 90 *Pruebas Heurísticas* 92

Figura 91 *Pruebas Heurísticas* 93

Resumen (Abstract)

El texto propone que para la Uniagustiniana una aplicación móvil que fomenta la instrucción en ciencias básicas, permitiendo clases puntuales y en corto plazo. El método actual para obtener asistencia académica se retrasa y se administra a través de un formulario de Google. Estudiantes de la carrera de Tecnología en Ingeniería de Software identificaron el potencial para potenciar la eficiencia de este método y perfeccionar el trabajo dentro de permanencia estudiantil. La aplicación integrará características como, incluido el alistamiento seguro de alumnos e instructores, verificación, manejo de perfiles y un mecanismo para organizar tutorías. Los estudiantes tendrán la opción de buscar y solicitar ayuda a los tutores, mientras que los tutores supervisarán su disponibilidad y mantendrán un diálogo directo a través de un sistema de mensajería interno. La implementación utiliza la metodología Mobile D, La innovación comprende una interfaz sencilla y fácil de usar, complementada con una autoevaluación que el estudiante le podrá dar al docente para que así misma permanencia pueda ver la evolución del proceso.

Palabra clave: *aplicación móvil, ciencias básicas, permanencia estudiantil, mobile D*

Abstract (resumen)

The text proposes that for Uniagustiniana a mobile application that promotes instruction in basic sciences, allowing punctual and short-term classes. The current method of obtaining academic assistance is delayed and administered through a Google form. Students of the Software Engineering Technology degree identified the potential to enhance the efficiency of this method and perfect the work within student permanence. The application will integrate features such as, including secure enrollment of students and instructors, verification, profile management and a mechanism for organizing tutorials. Students will have the option to seek and request help from tutors, while tutors will monitor their availability and maintain direct dialogue through an internal messaging system. The implementation uses the Mobile D methodology. The innovation includes a simple and easy-to-use interface, complemented with a self-assessment that the student can give to the teacher so that they can also see the evolution of the process.

Keywords: mobile application, basic sciences, student permanence, mobile D

Introducción

La Universidad Uniagustiniana, a través del área de permanencia estudiantil, ofrece un servicio de tutorías en ciencias básicas para apoyar el aprendizaje de sus estudiantes. Sin embargo, el proceso actual de agendamiento de estas tutorías presenta ciertas limitaciones en términos de eficiencia y rapidez, lo que afecta la experiencia del estudiante. Actualmente, el sistema de tutorías se gestiona mediante formularios de Google, cuyo procesamiento requiere la intervención de un funcionario que contacta al docente, y toma hasta tres días hábiles para coordinar la sesión solicitada. Este proceso se realiza de manera indirecta y resulta en una experiencia poco fluida para los estudiantes que necesitan reforzar sus conocimientos de manera oportuna.

En respuesta a este desafío, los estudiantes del programa de Tecnología en Desarrollo de Software de la Uniagustiniana han identificado la oportunidad de crear una aplicación móvil que optimice el proceso de agendamiento de tutorías, facilitando el acceso a estas y agilizando la comunicación entre estudiantes y docentes. Así, el **objetivo principal** de este proyecto es desarrollar una aplicación que permita una gestión directa, rápida y accesible de las tutorías en ciencias básicas. Mediante esta aplicación, los estudiantes podrán solicitar tutorías, seleccionar horarios y comunicarse con sus tutores de forma intuitiva y eficaz, promoviendo una experiencia de aprendizaje más dinámica y personalizada.

Desde una perspectiva académica y social, el desarrollo de esta aplicación tiene un valor significativo, ya que no solo introduce una solución tecnológica para optimizar el sistema de tutorías, sino que también fortalece los programas de apoyo estudiantil que son fundamentales para la retención y éxito académico de los estudiantes. La implementación de esta herramienta

puede servir como modelo para otras instituciones de educación superior, al demostrar cómo la tecnología puede integrarse en los programas de apoyo académico para reducir las barreras en el acceso a ayuda oportuna y promover el aprendizaje continuo.

Para abordar el desarrollo de la aplicación, emplea la **metodología Mobile D**, un enfoque ágil diseñado para la creación de aplicaciones móviles. Esta metodología se basa en el desarrollo iterativo y la retroalimentación constante de los usuarios, lo que permite ajustar y mejorar la aplicación conforme avanzan las fases del proyecto. Las etapas incluyen la investigación y comprensión de las necesidades de los usuarios, la definición de requisitos y funcionalidades clave, el diseño de prototipos y pruebas de usabilidad, y finalmente el desarrollo y lanzamiento de la aplicación con pruebas continuas para asegurar su estabilidad y usabilidad.

1. Datos informativos del proyecto

1.1 Título del proyecto

Desarrollo de una aplicación móvil como apoyo académico en ciencias básicas para el área de permanencia estudiantil de la universitaria Uniagustiniana

1.2 Síntesis

Desarrollar una aplicación móvil para el apoyo del aprendizaje de los estudiantes agustinianos la cual hará uso de una base datos para el almacenamiento de los usuarios (estudiantes) y docentes donde se pueda agendar tutorías para el refuerzo del aprendizaje, con procesos más ágiles y sencillos para el acceso oportuno a las tutorías para un refuerzo de los conocimientos.

2 Planificación del proyecto

2.1 Planteamiento del problema

Las tutorías son un servicio que ofrece la universidad a través del área de permanencia al cual los estudiantes pueden acceder de manera gratuita para reforzar conocimientos o mejorar en sus dificultades académicas. Para acceder a las tutorías se debe pasar por un formulario de Google, lo cual esa información será redirigida por un funcionario de la universidad en el área de permanencia hacia el docente el cual se comunicará en los siguientes tres días hábiles con el estudiante para agendar el día oportuno de la tutoría.

Estudiantes del programa de Tecnología de Desarrollo de Software realizaron un diagnóstico del proceso de agendamiento de las tutorías evidenciando la oportunidad de hacerla más ágil ya que el área permanencia cuenta con una plataforma (Adviser) que no es las más eficiente y poco usada por lo que el procedimiento existente es más demorado.

Los estudiantes del programa de Tecnología de Desarrollo de Software se reunieron con la directora del área de permanencia con el fin de establecer las necesidades y requerimientos para que el proceso de agendamiento de tutorías sea más ágil y eficiente.

Posteriormente los estudiantes del programa de Tecnología de Desarrollo de Software analizaron las necesidades presentadas por el área de permanencia con el fin de establecer un plan de trabajo para llevar a cabo dicho desarrollo.

2.2 Objetivos del proyecto

2.2.1 Objetivo general

Desarrollar una aplicación móvil para el área de permanencia estudiantil con el fin de brindar apoyo a los estudiantes de la universitaria Uniagustiniana en ciencias básicas, agilizando el método de agendamiento de las tutorías.

2.2.2 Objetivos Específicos

Analizar las necesidades de los estudiantes cuando buscan tutorías para poder identificar qué aspectos clave deben ser cubiertos para mejorar su aprendizaje en ciencias básicas.

Definir los requisitos que debe tener la aplicación, tanto en funciones como en aspectos técnicos, asegurándonos de que realmente cumpla con lo que los estudiantes necesitan.

Programar e integrar una base de datos para gestionar y sincronizar datos de manera segura y eficiente entre la aplicación y el servidor, asegurando información actualizada y accesible

Probar la aplicación a fondo con pruebas unitarias y de interfaz para asegurarnos de que todo funcione bien y ofrecer una herramienta de calidad a los estudiantes.

2.3 Alcance de la solución

El proyecto se llevará a cabo con el fin de ayudar a los estudiantes de la universitaria Uniagustiniana con el tema de las asignaciones de las tutorías, la solución se basa en la creación de una aplicación móvil que estará dirigida a la parte de permanencia estudiantil de la universidad

Esta implementación móvil estará enfocada en agilizar los pasos que ya se tienen haciendo un proceso más rápido y de mejor rendimiento para ambas partes, que es fundamental para asegurar que los estudiantes puedan mantenerse al día con sus materias en ciencias básicas y avanzar en sus estudios con tranquilidad, además el estudiante podrá darle una calificación al docente que le dictó la tutoría, esto con el fin de hacerle un seguimiento al proceso y así mismo permanencia pueda determinar en qué parte se puede seguir mejorando .

el enfoqué de la aplicación es que los estudiantes se sientan más familiarizados con las tutorías, de igual manera para el docente poder tener más contacto con el estudiante para poder abordar de manera clara las áreas que requieren refuerzo, contribuyendo a la adaptación y el progreso académico dentro de la institución.

2.4 Metodología de desarrollo de software

Para el desarrollo de la aplicación móvil, se utilizó la metodología ágil de desarrollo de aplicaciones móviles Mobile-D, esto debido a su enfoque práctico y ágil destinado para grupos pequeños de desarrollo. “El objetivo de esta metodología es conseguir ciclos de desarrollo muy rápidos en equipos muy pequeños” (Syntonize, 2021, p. 1)

Esta metodología se compone de cinco fases que definen los pasos a seguir para el desarrollo e implementación de estas, fases las cuales serán detalladas a continuación.

2.4.1 Fase de exploración

Esta fase inicial es la encargada de definir las características principales del proyecto como los usuarios y el alcance que este tendrá. “Las tareas asociadas a esta fase incluyen el establecimiento del cliente (los clientes que toman parte activa en el proceso de desarrollo), la planificación inicial del proyecto y los requisitos de recogida, y el establecimiento de procesos” (Balaguera, 2013, p. 118)

El dúo en esta fase investigó el proceso actual del agendamiento de las tutorías, observando todo su proceso para poder identificar los puntos débiles de este, y así poder definir las necesidades, características y alcance que debe tener la aplicación móvil.

2.4.2 Fase de inicialización

La fase de inicialización es fundamental ya que en esta se definen los recursos necesarios para el desarrollo de la aplicación móvil y las etapas posteriores. En esta se establecen los componentes de hardware y software, así como el almacenamiento requerido para la solución móvil propuesta. “Esta fase se divide en cuatro etapas: la puesta en marcha del proyecto, la planificación inicial, el día de prueba y el día de salida” (Balaguera, 2013, p. 119).

Para esta fase, el dúo empezó con la realización del documento del proyecto, además definió los recursos necesarios para el desarrollo de la aplicación móvil, optando por herramientas como Visual Studio, debido a su integración con el lenguaje de programación Flutter, y Firebase para la gestión de la base de datos, lo cual permite un desarrollo óptimo y sin complicaciones.

2.4.3 Fase de producción

En esta fase como indica su nombre se empieza con el desarrollo de la aplicación móvil, donde poco a poco está tomará forma, se empezará con la creación de módulos de manera separada los cuales serán unificados. Según Balaguera (2013) en la fase de producción:

> "Se repite la programación de tres días (planificación, trabajo, liberación) de manera iterativa hasta implementar todas las funcionalidades. Primero se planifica la iteración de trabajo en términos de requisitos y tareas a realizar. Se preparan las pruebas de la iteración de antemano. Las tareas se llevarán a cabo durante el día de trabajo, desarrollando e integrando el código con los repositorios existentes. Durante el último día se lleva a cabo la integración del sistema (en caso de que estuvieran trabajando varios equipos de forma independiente) seguida de las pruebas de aceptación." (p. 119).

Para el desarrollo de esta fase el dúo dividió el trabajo en módulos, priorizando los más complejos como el módulo de chat o el módulo de agendamiento de tutorías con un apartado visual básico, todo esto siendo almacenados en una rama de GitHub donde estos serán unidos dando forma a la aplicación móvil, para por último pulir y mejorar el apartado gráfico.

2.4.4 *Fase de estabilización*

Una vez terminada la parte de producción se empieza con la fase de estabilización, un paso muy importante donde se realizan las últimas integraciones buscando que el software funcione como se espera. "se realizan las acciones de integración para enganchar los posibles módulos separados en una única aplicación." (Syntonize, 2021, p. 1)

Iniciada esta fase el dúo se encargó de culminar con la documentación y unificar completamente los módulos e implementar la base de datos, esto dando como resultado un backend, frontend y sus funcionalidades de almacenamiento, escritura y lectura unificados y

operativos dando paso a la identificación de errores y bugs que procederán a ser solucionados y eliminados.

2.4.5 Fase de prueba

La fase de pruebas es la última en la metodología Mobile-D, esta es la encargada de la generación de la versión estable de la aplicación móvil, donde se corregirán los últimos errores y bugs presentes para así poder realizar la entrega final, esto dando final al desarrollo y proceso de la metodología.

> "La última fase (prueba y reparación del sistema) tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema. El producto terminado e integrado se prueba con los requisitos del cliente y se eliminan todos los defectos encontrados." (Balaguera, 2013, p. 119)

Para la última fase el dúo se encargó de mejorar la estabilidad y corregir errores que afecten la funcionalidad correcta de la aplicación móvil, todo esto con el fin de tener la versión final de esta para su demostración.

3 Especificación de requisitos de software (IEEE830)

3.1 Funcionalidad del producto

3.1.1 Registro y Gestión de Usuarios

Registro de Estudiantes y Docentes:

Los estudiantes y docentes podrán registrarse en la aplicación creando un perfil con la información básica, como lo es su nombre, correo electrónico institucional y el rol (estudiante o docente)

Inicio de Sesión:

Para proteger la información de los usuarios, la aplicación contará con un sistema seguro de inicio de sesión.

Gestión de Perfiles:

Los usuarios tendrán la visualización de su información y un apartado donde podrán actualizar su información como cambiar su contraseña.

3.1.2 Agendamiento de Tutorías

Solicitud de Tutorías:

Los estudiantes tendrán acceso a una interfaz intuitiva donde podrán buscar tutores en ciencias básicas, podrán filtrar las opciones por materia, lo que les permitirá encontrar una opción más fácil. Una vez seleccionada la tutoría, podrán confirmar su solicitud en pocos pasos

Comunicación Docente estudiante:

Los estudiantes y los docentes cuentan con un apartado que es sobre un chat en tiempo real, facilitando y agilizando la comunicación entre ellos mismos, haciendo que la parte del agendamiento sea más fácil y flexible

Vista Detallada de Tutorías:

Al momento que la tutoría quede asignada, los estudiantes podrán revisar información adicional como el tema específico Esto ayudará a que puedan tomar decisiones más informadas al momento de agendar.

3.1.3 Gestión de Tutorías

Eliminación de tutorías:

Para este proceso el único que podrá eliminar o modificar la tutoría será por medio del administrador, esto haciéndolo por medio de correo solicitando esa solicitud.

Historial de Tutorías:

La aplicación ofrecerá un registro detallado de todas las tutorías tomadas, esto permitirá a los estudiantes, administrativos y docentes revisar qué temas han trabajado y su progreso con la aplicación.

3.1.4 Comunicación

Mensajería Interna:

La plataforma incluirá una herramienta de chat de forma directa para que estudiantes y docentes puedan comunicarse fácilmente. Esto será útil para coordinar detalles previos a una tutoría, tales como horario y tema que se desee repasar.

3.1.5 Base de Datos

Gestión de Datos:

Toda la información de los usuarios, tutorías y registros será almacenada de manera segura en una base de datos confiable.

3.1.6 Interfaz de Usuario

Diseño Intuitivo:

La aplicación tendrá un diseño para ser fácil de usar, su interfaz permitirá a estudiantes, docentes y administrativos navegar sin ninguna complicación, facilitando el acceso a las funcionalidades principales.

3.2 Características de los Usuarios

3.2 Especificaciones de los Usuarios

Estudiantes Alumnos

Requerimientos: Acceso a tutorías en materias básicas, una comunicación sencilla con los docentes y estudiantes para programar la tutoría.

Preferencias: Una interfaz sencilla navegar sin problemas, búsqueda y filtrado, notificación en las conversaciones con los profesores.

Docentes

Requerimientos: Flexibilidad para establecer horarios de presencia, capacidad para interactuar con los estudiantes y aprovechar la experiencia de la sesión mediante un chat.

Administrador: Visualización de las tutorías, modificación de las tutorías, creación de un documento PDF de un mes sobre la experiencia de las tutorías, incorporación de nuevos profesores y supresión de estudiantes.

3.3 Requisitos funcionales

Tabla 1

Requisitos funcionales

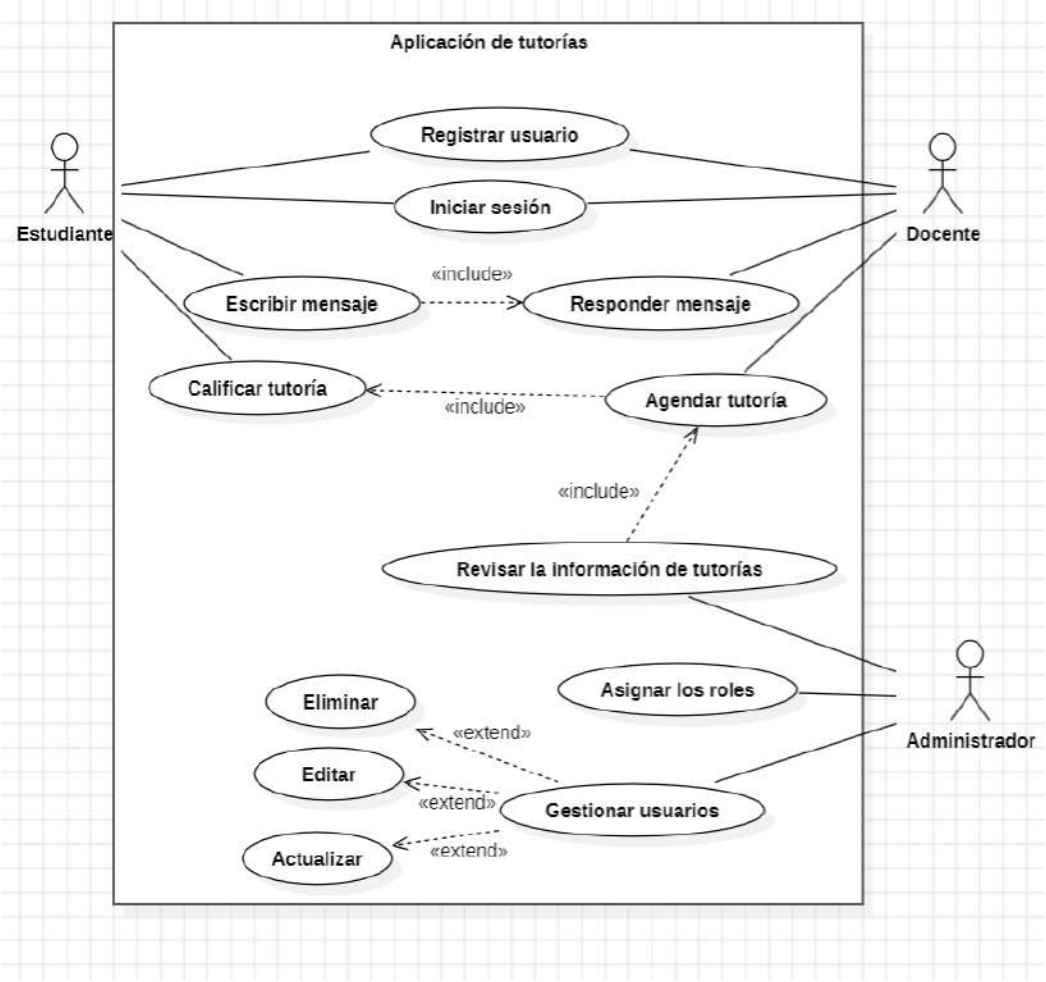
ID-RQF	Módulo/Funcionalidad	Descripción requisito	Actor	Prioridad
01	Autenticación	El sistema debe tener un módulo de login para que los usuarios puedan autenticarse y acceder a la aplicación.	Usuario	Alta
02	Registro de Usuarios	El sistema debe permitir que los usuarios (nombre, correo, contraseña) se registren para acceder a la aplicación.	Usuario	Alta
03	Lista de Docentes	El sistema debe mostrar la lista de docentes disponibles para tutorías a los usuarios con rol de estudiante.	Estudiante	Alta
04	Chat en Tiempo Real	El sistema debe contar con un chat en tiempo real para que los estudiantes interactúen con los docentes.	Estudiante, Docente	Alta
05	Gestión de Usuarios	El administrador puede editar la información de todas las cuentas registradas	Administrador	Alta
06	Gestión de Usuarios	El administrador puede eliminar la información de todas las cuentas registradas	Administrador	Alta
07	Gestión de Usuarios	El administrador puede debe actualizar la información de todas las cuentas registradas	Administrador	Alta
08	Gestión de Usuarios	El administrador puede agregar una nueva cuenta al sistema.	Administrador	Media
09	Información de Docentes	El sistema debe permitir a los estudiantes visualizar la información básica de los docentes.	Estudiante	Alta
10	Información de los estudiantes	Los docentes deben poder ver la información básica de los estudiantes (nombre, correo, carrera, semestre).	Docente	Media
11	Asignación de Tutorías	El administrador puede consultar las tutorías asignadas con información de horario, estudiante y docente.	Administrador	Alta
12	Asignación de Tutorías	El docente podrá asignar una tutoría al sistema especificando horario y al estudiante a quien va dirigida.	Docente	Alta
13	Edición de Perfil	Los usuarios pueden modificar su información personal en cualquier momento.	Usuario	Media

Nota: Autoría propia (2024)

3.4 Diagrama de casos de uso

Figura 1

Diagrama de casos de uso



Nota: Autoría propia (2024)

3.5 Especificaciones de los casos de uso

Tabla 2

Tabla de registro

<table>
<tr><td>Nombre:</td><td colspan="2">Gestionar Registro de Usuarios</td></tr>
<tr><td>Autor:</td><td colspan="2">Manuel Alejandro Lozano Hernandez - Cristobal Fernando pérez chavarro</td></tr>
<tr><td>Fecha:</td><td colspan="2">07/09/2024</td></tr>
<tr><td colspan="3">Descripción:

Permitir que los usuarios se registren en la aplicación proporcionando sus credenciales.</td></tr>
<tr><td colspan="3">Actores: Estudiante, Docente</td></tr>
<tr><td colspan="3">Precondiciones:
El usuario debe tener acceso a la aplicación y no estar registrado previamente.</td></tr>
<tr><td>Flujo Normal:

1. El usuario accede a la pantalla de inicio.
2. Selecciona la opción "Registrarse".
3. Ingresa sus datos (nombre, correo, contraseña, rol).
4. El sistema valida que el correo no esté en uso.
5. El sistema almacena la información.
6. El usuario es redirigido a su panel.</td><td colspan="2">Flujo Alternativo:

4A. Si el correo ya está en uso, el sistema muestra un mensaje de error y solicita un nuevo correo o proceder con la recuperación de contraseña.</td></tr>
<tr><td colspan="3">Poscondiciones: El usuario queda registrado y puede iniciar sesión la próxima vez.</td></tr>
<tr><td colspan="3">Prioridad: Alta</td></tr>
</table>

Nota: Autoría propia (2024)

Tabla 3

Tabla de inicio de sesión

Nombre:	**Gestionar Autenticación de Usuarios**
Autor:	***Manuel Alejandro Lozano Hernandez*** ***- Cristobal Fernando pérez chavarro***
Fecha:	07/09/2024
Descripción: Permitir que los usuarios inicien sesión en la aplicación con sus credenciales registradas.	
Actores: Estudiante, Docente, Administrador	
Precondiciones: El estudiante debe estar autenticado y haber docentes disponibles en la base de datos.	
Flujo Normal: 1. El usuario accede a la pantalla de inicio. 2. Selecciona la opción "Iniciar Sesión". 3. Ingresa su correo y contraseña. 4. El sistema valida las credenciales. 5. El usuario accede a la aplicación.	Flujo Alternativo: 3A. Si el usuario no recuerda la contraseña, selecciona la opción de recuperación mediante correo.
Poscondiciones: El usuario accede a su panel, donde puede gestionar sus tutorías y modificar su información personal.	
Prioridad: Alta	

Nota: Autoría propia (2024)

Tabla 4

Tabla de solicitud

Nombre:	Solicitud de Tutoría por el Estudiante
Autor:	***Manuel Alejandro Lozano Hernandez*** ***- Cristobal Fernando pérez chavarro***
Fecha:	07/09/2024
Descripción: Los estudiantes pueden solicitar tutorías seleccionando al docente disponible	
Actores: Estudiante, Docente	
Precondiciones: El estudiante debe estar autenticado y haber docentes disponibles en la base de datos.	
Flujo Normal: 1. El estudiante accede al módulo de tutorías. 2. Selecciona un docente. 3. oprime el botón de enviar solicitud 4. El sistema envía una solicitud al docente. 5. El docente se comunica por medio del chat con el estudiante para cuadrar horarios y necesidades 1. El estudiante accede al módulo de tutorías. 2. Selecciona un docente disponible. 3. Oprime el botón de enviar solicitud. 4. El sistema envía la solicitud al docente. 5. El docente se comunica mediante el chat para cuadrar horarios y detalles de la tutoría. 6. El sistema confirma la tutoría al estudiante.	Flujo Alternativo: 4A. la solicitud falla y el sistema muestra un mensaje recomendando internar realizar la solicitud de nuevo. 4A. Si la solicitud falla (por problemas de conexión o docente no disponible), el sistema muestra un mensaje recomendando intentar realizar la solicitud nuevamente.
Poscondiciones: La tutoría queda agendada. La tutoría queda registrada en el sistema y el estudiante recibe una confirmación visual en su panel o mediante una notificación.	
Prioridad: Alta	

Nota: Autoría propia (2024)

Tabla 5

Tabla del chat

Nombre:	Chat en Tiempo Real entre Estudiante y Docente
Autor:	***Manuel Alejandro Lozano Hernandez*** **- *Cristobal Fernando pérez chavarro***
Fecha:	07/09/2024
Descripción: El estudiante y el docente pueden comunicarse en tiempo real para resolver dudas sobre la tutoría.	
Actores: Estudiante, Docente	
Precondiciones: Ambos actores deben estar autenticados y haber una solicitud de tutoría entre ellos.	
Flujo Normal: 1. El docente se comunica con el estudiante al recibir la solicitud de tutoría 2.El estudiante accede al chat desde su panel de chats. 3. El docente responde en tiempo real. 4. El sistema asegura la entrega instantánea de los mensajes.	Flujo Alternativo: 3A. Si el docente no está disponible, el sistema guarda el mensaje para ser respondido más tarde.
Poscondiciones: La conversación queda registrada para futuras consultas.	
Prioridad: Alta	

Nota: Autoría propia (2024)

Tabla 6

Tabla de tutorías

<table>
<tr><td>Nombre:</td><td colspan="2">Revisión de Tutorías Asignadas y Calificación</td></tr>
<tr><td>Autor:</td><td colspan="2">Manuel Alejandro Lozano Hernandez - Cristobal Fernando pérez chavarro</td></tr>
<tr><td>Fecha:</td><td colspan="2">07/09/2024</td></tr>
<tr><td colspan="3">Descripción:

El administrador puede revisar las tutorías asignadas a los estudiantes, incluyendo la información de cada tutoría (horario, docente, estudiante) y la calificación dada por el estudiante.</td></tr>
<tr><td colspan="3">Actores: Administrador</td></tr>
<tr><td colspan="3">Precondiciones: El administrador debe estar autenticado en el sistema y haber tutorías previamente asignadas.</td></tr>
<tr><td colspan="2">Flujo Normal:

1. El administrador accede al módulo de revisión de tutorías.
2. El sistema muestra una lista de tutorías asignadas con su respectiva información: estudiante, docente, fecha y hora.
3. El administrador selecciona una tutoría específica para revisar los detalles.
4. El sistema muestra la calificación dejada por el estudiante sobre la tutoría.</td><td>Flujo Alternativo:

3A. Si no hay tutorías asignadas, el sistema notifica que no existen registros para mostrar.</td></tr>
<tr><td colspan="3">Poscondiciones: El administrador ha revisado la información de las tutorías asignadas y puede usar los datos para análisis o tomar decisiones.</td></tr>
<tr><td colspan="3">Prioridad: Alta</td></tr>
</table>

Nota: Autoría propia (2024)

3.6 Restricciones y Atributos de Calidad

Tabla 7

Tabla de Atributos de calidad.

ID-RQNF	Atributo de Calidad	Módulo/Funcionalidad	Prioridad
01	El sistema debe responder en menos de 10 segundos a las acciones del usuario.	Tiempo de Respuesta	Alta
02	El sistema debe manejar hasta 20 solicitudes por minuto sin afectar el rendimiento.	Rendimiento	Alta
03	La interfaz debe ser intuitiva y de fácil uso para todos los usuarios.	Usabilidad	Alta
04	El sistema debe ejecutarse en dispositivos Android (7 o superior) y iOS (11 o superior).	Compatibilidad	Alta
05	La aplicación debe estar disponible 24/7 durante el semestre universitario.	Disponibilidad	Alta
06	El sistema debe tener un modo oscuro para mejorar la experiencia del usuario.	Modo oscuro	Media
07	Los mensajes entre estudiantes y docentes deben ser instantáneos.	Entrega instantánea de mensajes	Alta
ID-RQNF	Restricción	Módulo/Funcionalidad	Prioridad
01	El sistema debe ser desarrollado en Dart para Flutter.	Lenguaje de Desarrollo	Alta
02	Solo el administrador podrá modificar los roles de los usuarios.	Rol de Administrador	Alta

Nota: Autoría propia (2024)

4 Diseño del software (ISO-12207-1) 36

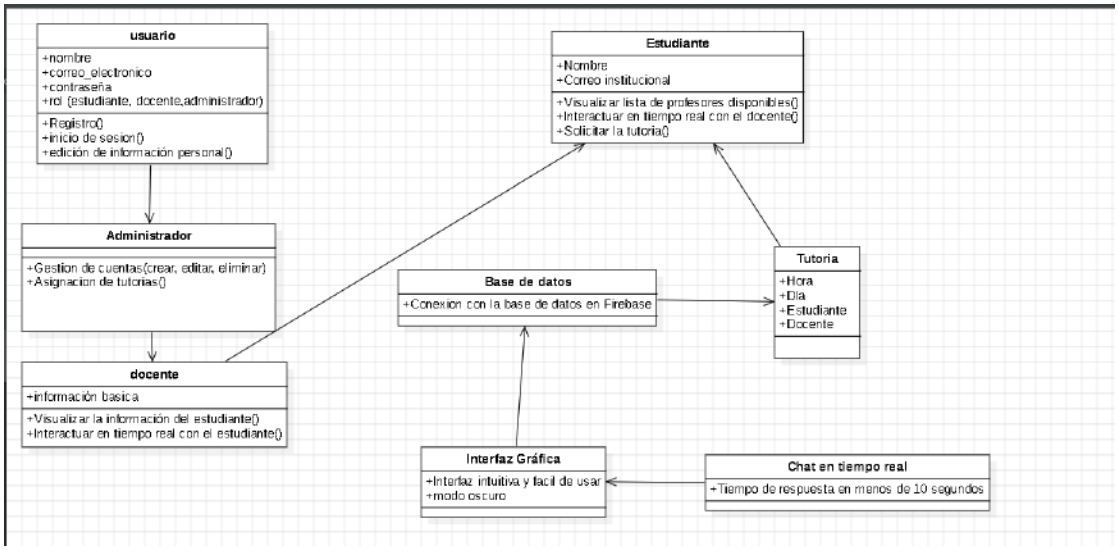
4.1 Diseño detallado del software

4.1.1 Diagrama de clases

Muestra las clases del sistema, sus atributos y métodos, así como las relaciones entre ellas.

Figura 2

Diagrama de clases



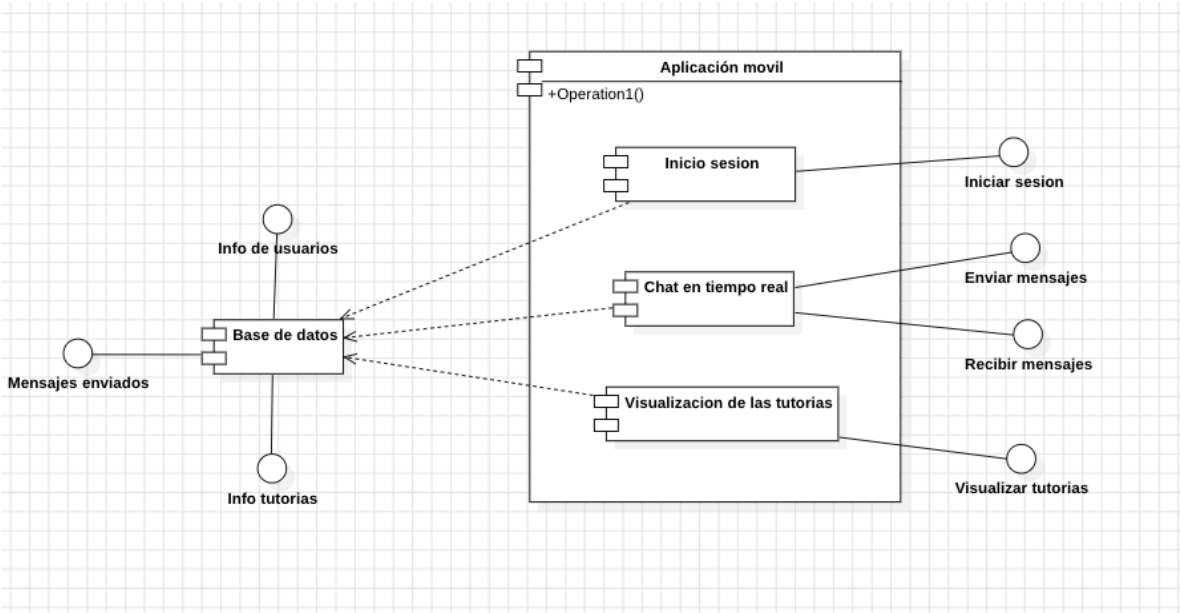
Nota: Autoría propia (2024)

4.1.2 *Diagrama de componentes*

Describe los módulos o partes principales de la aplicación y cómo interactúan entre sí, ayudando a entender la arquitectura general del sistema.

Figura 3

Diagrama de componentes



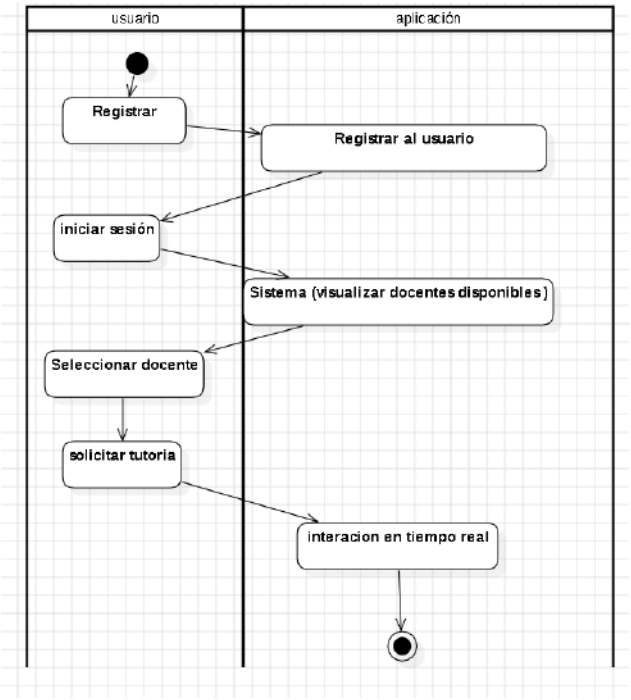
Nota: Autoría propia (2024)

4.1.3 Diagrama de actividades

Representa el flujo de trabajo o los pasos de los procesos en la aplicación, mostrando el orden y la lógica detrás de cada actividad.

Figura 4

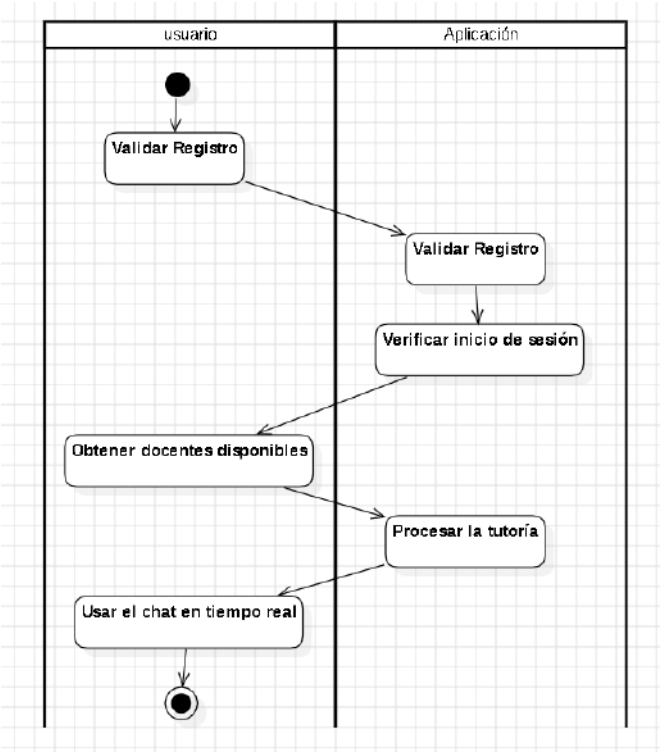
Diagrama de actividades



Nota: Autoría propia (2024)

Figura 5

Diagrama de actividades



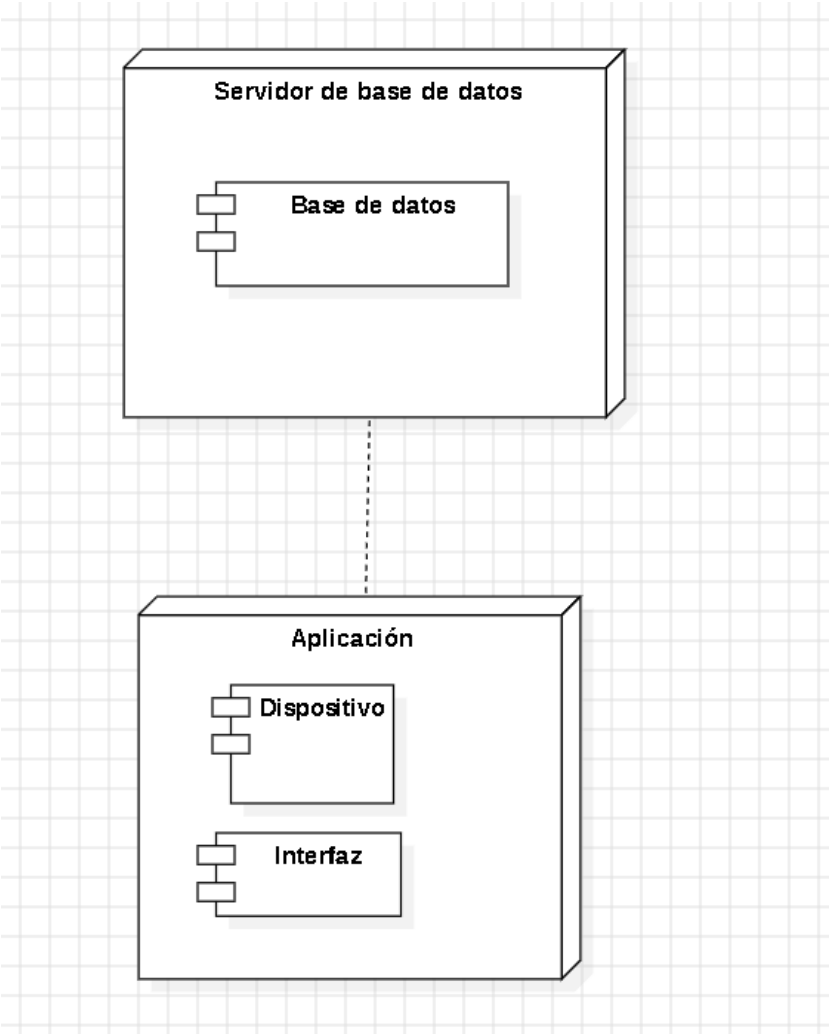
Nota: Autoría propia (2024)

4.1.4 Diagrama de despliegue

Muestra la distribución de los componentes de software en el hardware (como servidores y dispositivos móviles) y cómo estos interactúan, importante para planificar la infraestructura y el rendimiento de la aplicación.

Figura 6

Diagrama de Despliegue



Nota: Autoría propia (2024)

4.2 Diseño de la Interfaz

4.2.1 Interfaz de Usuario (Hardware, Software y Comunicaciones)

La Interfaz de Usuario es el punto de interacción directa entre el usuario y la aplicación, y su diseño es fundamental para asegurar una experiencia de usuario intuitiva y agradable. En la sección de Diseño de la Interfaz del proyecto, se abordan los aspectos de hardware, software y comunicaciones necesarios para crear una interfaz accesible y funcional.

Figura 7

Interfaz de Usuario



Nota: Autoría propia (2024)

Figura 8

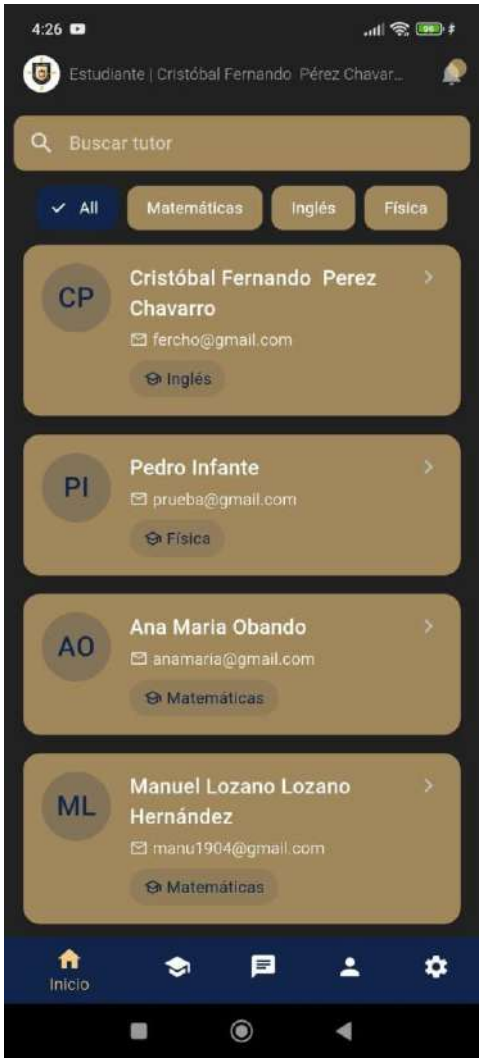
Interfaz de Usuario



Nota: Autoría propia (2024)

Figura 9

Interfaz de Usuario



Nota: Autoría propia (2024)

Figura 10

Interfaz de Usuario



Nota: Autoría propia (2024)

Figura 11

Interfaz de Usuario



Nota: Autoría propia (2024)

Figura 12

Interfaz de Usuario



Nota: Autoría propia (2024)

Figura 13

Interfaz de Usuario



Nota: Autoría propia (2024)

Figura 14

Interfaz de Usuario



Nota: Autoría propia (2024)

Figura 15

Interfaz de Usuario



Nota: Autoría propia (2024)

Figura 16

Interfaz de Usuario



Nota: Autoría propia (2024)

Figura 17

Interfaz de Usuario



Nota: Autoría propia (2024)

4.2.2 Interfaces de Entrada (Formularios, Log in)

Figura 18

Interfaz de Entrada



Nota: Autoría propia (2024)

Figura 19

Interfaz de Entrada



Nota: Autoría propia (2024)

Figura 20

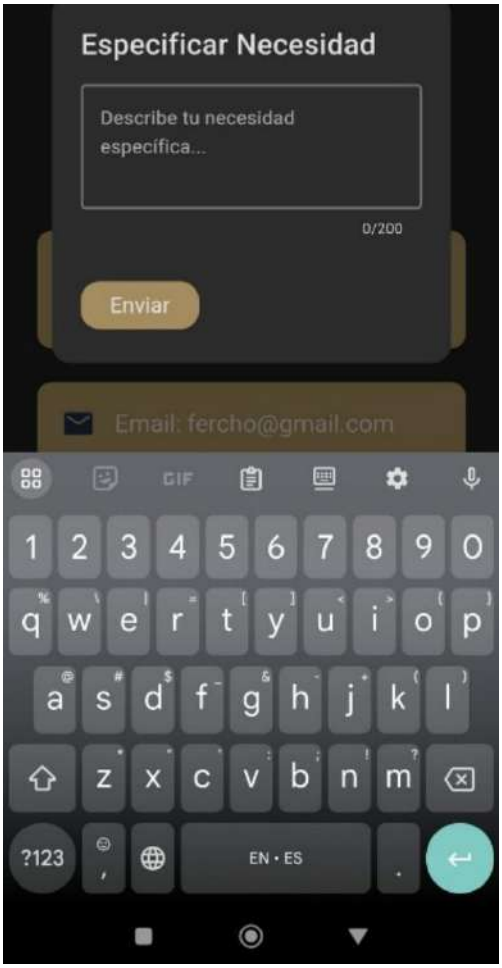
Interfaz de Entrada



Nota: Autoría propia (2024)

Figura 21

Interfaz de Entrada



Nota: Autoría propia (2024)

Figura 22

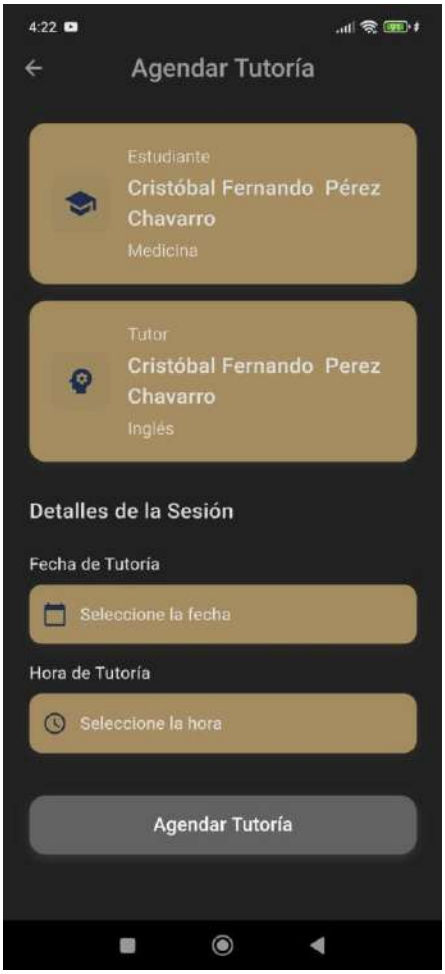
Interfaz de Entrada



Nota: Autoría propia (2024)

Figura 23

Interfaz de Entrada



Nota: Autoría propia (2024)

Figura 24

Interfaz de Entrada



Nota: Autoría propia (2024)

Figura 25

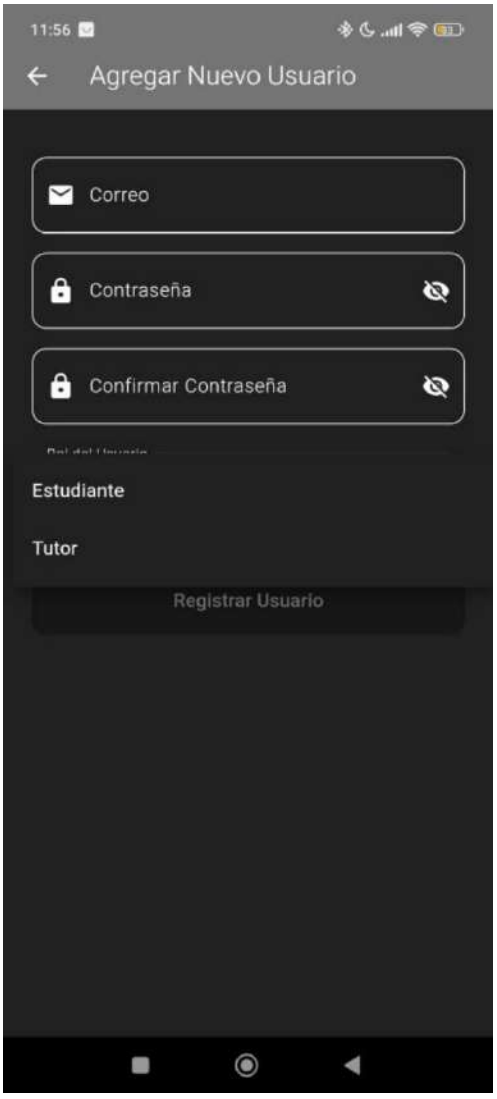
Interfaz de Entrada



Nota: Autoría propia (2024)

Figura 26

Interfaz de Entrada



Nota: Autoría propia (2024)

Figura 27

Interfaz de Entrada



Nota: Autoría propia (2024)

4.2.3 Interfaces de Salida (Reportes, Consultas, Impresiones)

Figura 28

Interfaz de Salida



Nota: Autoría propia (2024)

Figura 29

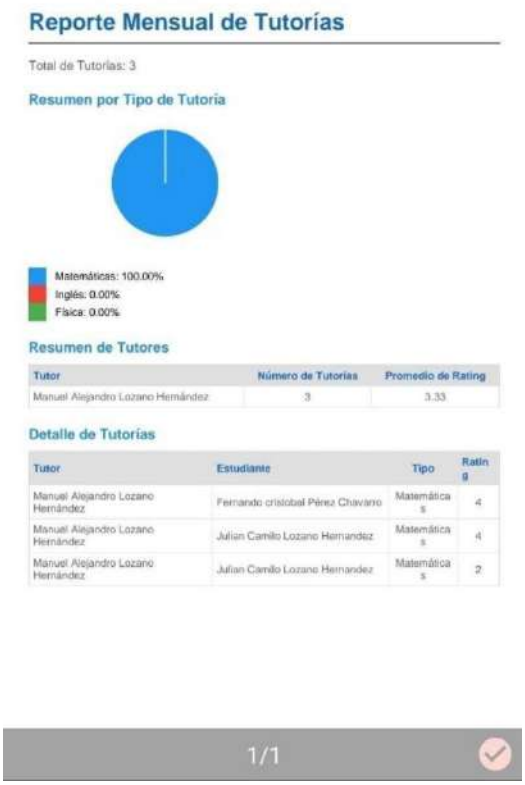
Interfaz de Salida



Nota: Autoría propia (2024)

Figura 30

Interfaz de Salida



Nota: Autoría propia (2024)

Figura 31

Interfaz de Salida



Nota: Autoría propia (2024)

Figura 32

Interfaz de Salida



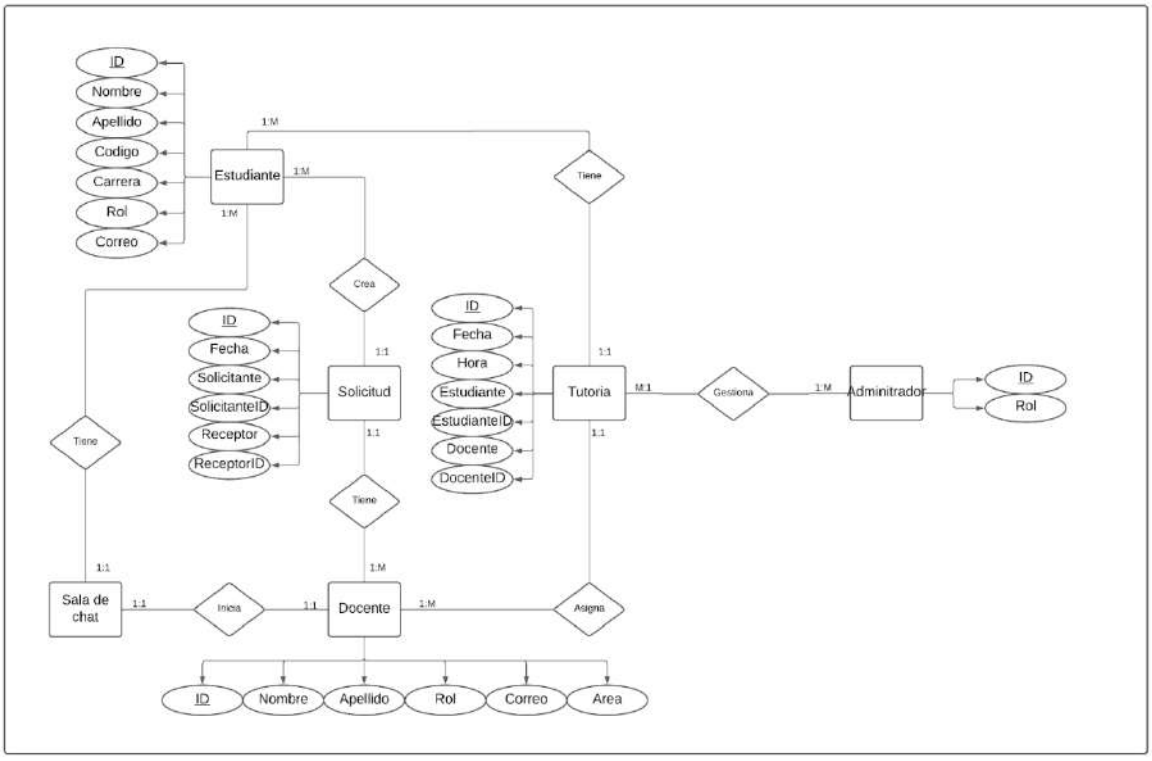
Nota: Autoría propia (2024)

4.3 Diseño del modelo de datos / Persistencia (Modelo E-R)

La Base de Datos es un componente central en el desarrollo de esta aplicación, ya que gestiona el almacenamiento, organización y acceso a la información de usuarios, tutorías y comunicación entre estudiantes y docentes. Su diseño está orientado a garantizar la seguridad, accesibilidad y eficiencia en el manejo de datos esenciales para el funcionamiento de la aplicación.

Figura 33

Diagrama de la Base de datos



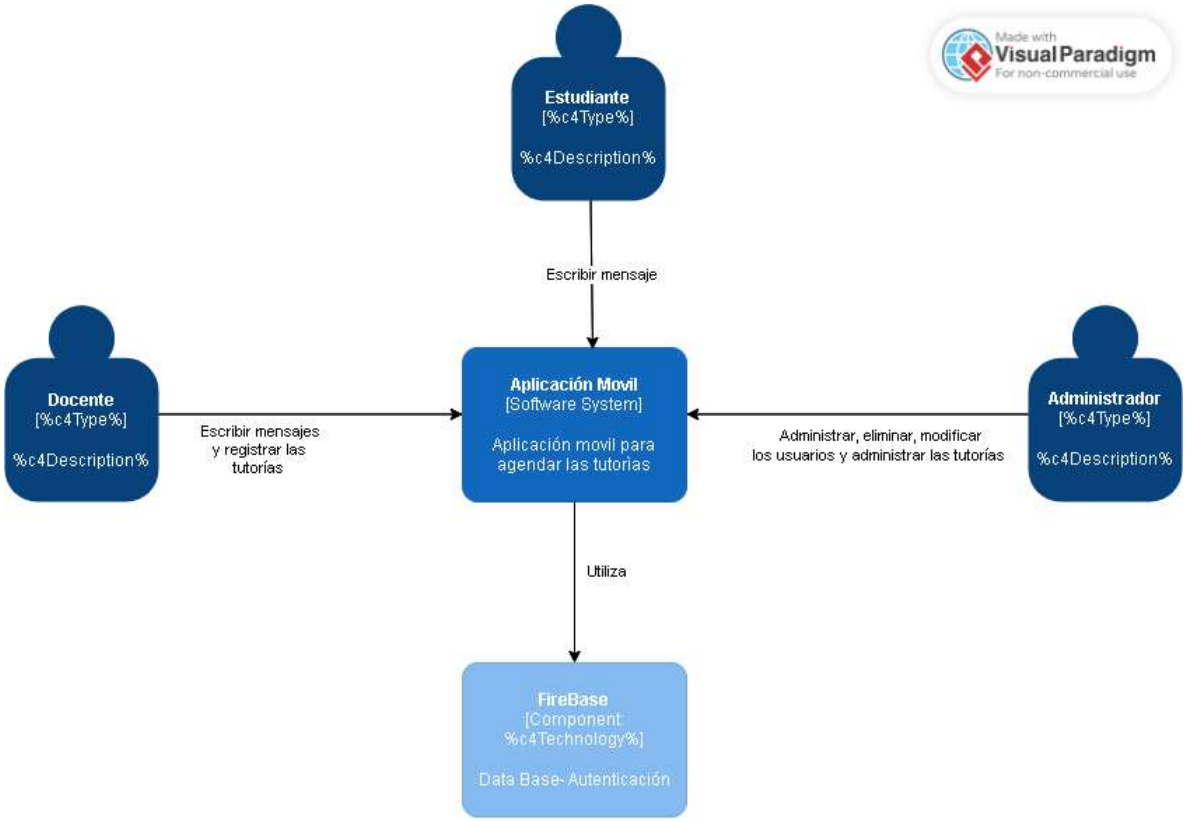
Nota: Autoría propia (2024)

4.4 Diseño de la Arquitectura de software (Modelo C4,nivel 1, nivel 2 y 3)

Este diagrama proporciona una visión general del sistema y su relación con los usuarios y otros sistemas externos. En el caso de la aplicación para tutorías, el diagrama de contexto mostraría a la aplicación móvil como el sistema central, en interacción con los **usuarios** (estudiantes y docentes) y el sistema de autenticación de la universidad.

Figura 34

Diagrama c4 Nivel 1



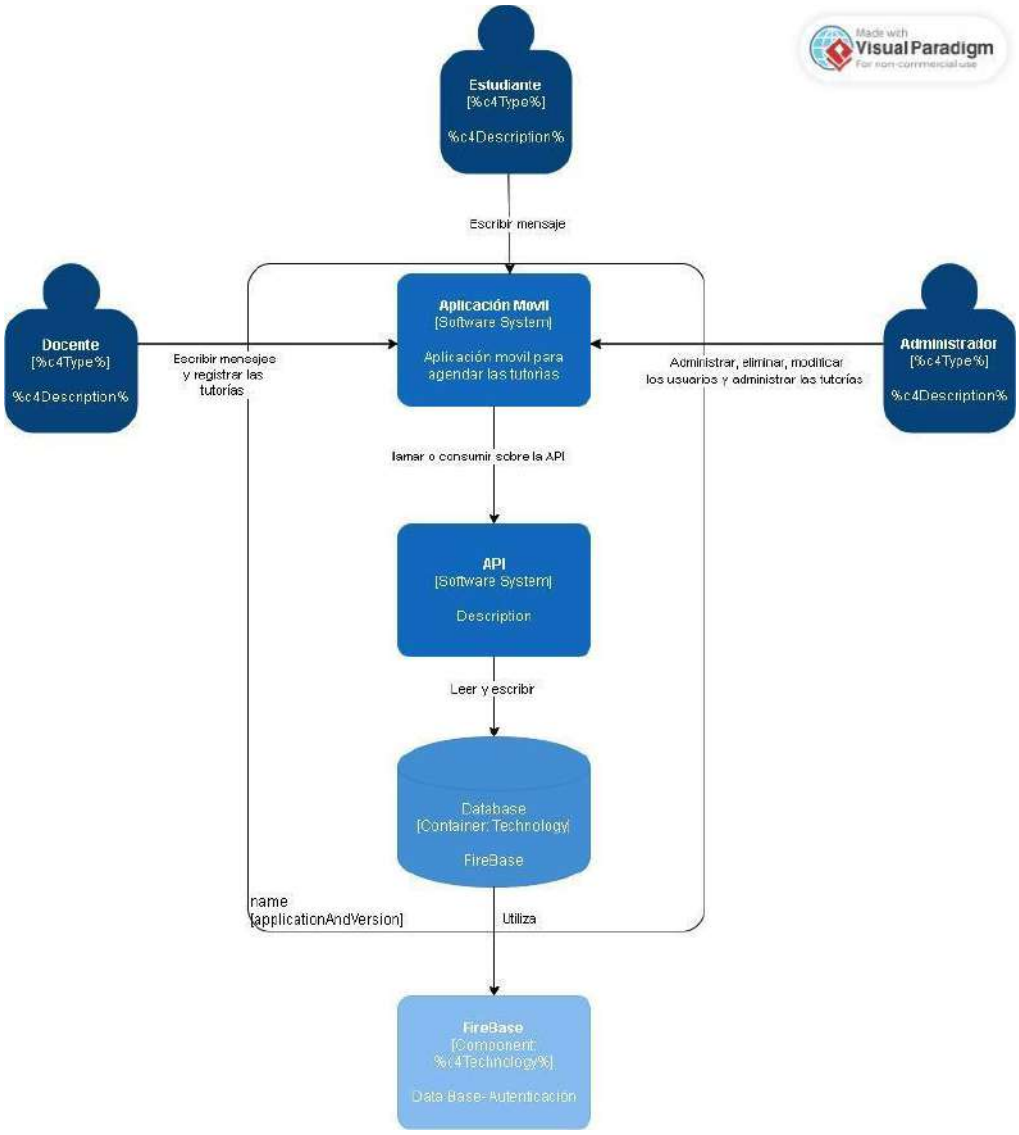
Nota: Autoría propia (2024)

4.5 Diagrama de contenedores

Aquí se detallan los diferentes “contenedores” o partes principales del sistema y sus interacciones.

Figura 35

Diagrama c4 Nivel 2



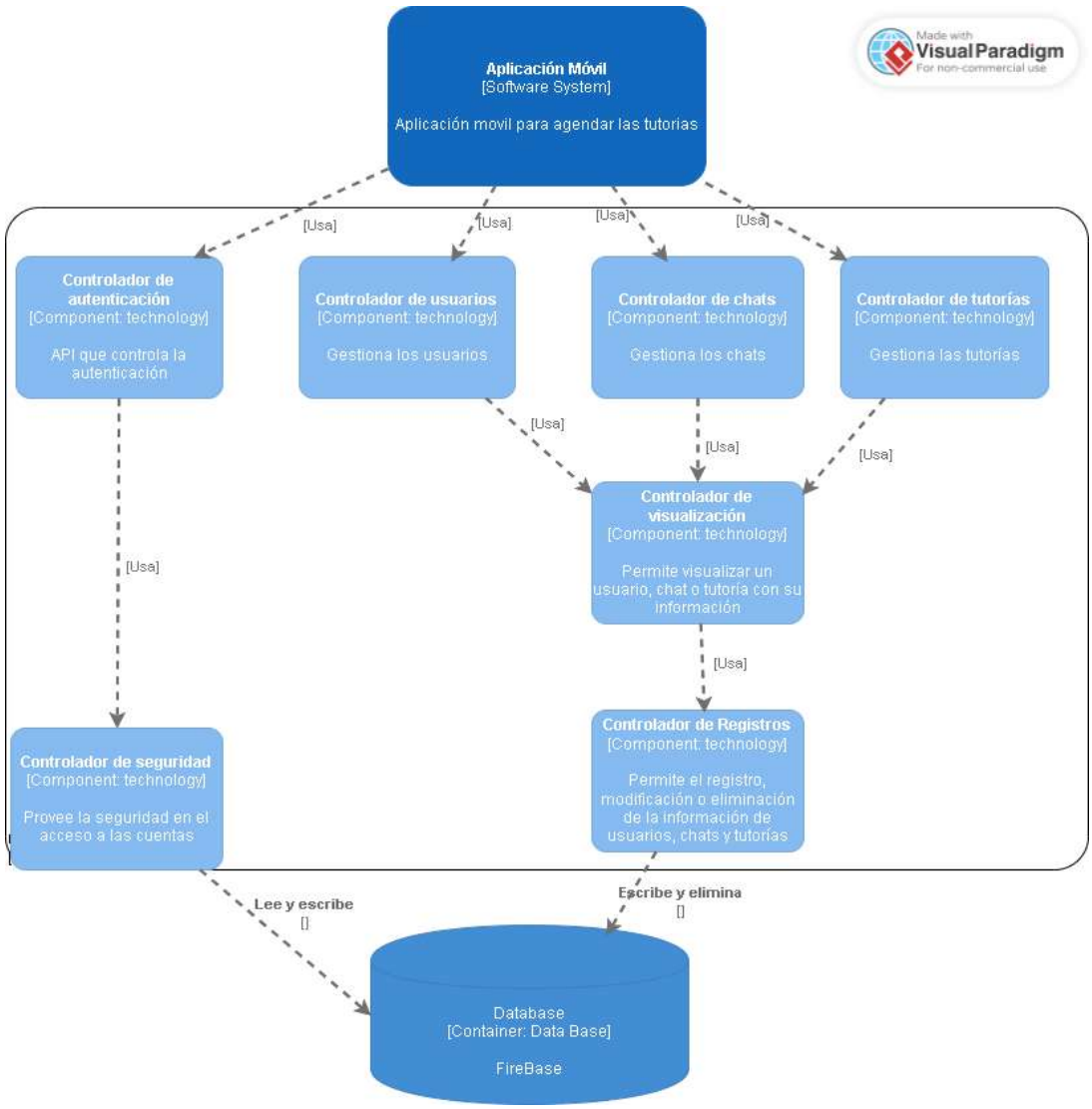
Nota: Autoría propia (2024)

4.6 Diagrama de componentes

Este nivel descompone cada contenedor en sus componentes específicos y muestra cómo trabajan juntos para ofrecer las funcionalidades de la aplicación.

Figura 36

Diagrama c4 Nivel 3



Nota: Autoría propia (2024)

5 Implementación

5.1 Herramientas utilizadas en el desarrollo del proyecto

Para el desarrollo de la aplicación móvil, se han utilizado herramientas que permitan un fácil y óptimo desarrollo, dando como fin un rendimiento óptimo y una experiencia de usuario de calidad.

Lenguaje de desarrollo y entorno de trabajo:

Como lenguaje de programación para la elaboración de la aplicación móvil se utilizó Dart junto con el framework Flutter. El desarrollo se realizó en Visual Studio Code, por su reconocida versatilidad y gran gama de extensiones para el desarrollo.

Base de datos y servicios en la nube:

Se implementaron los servicios de firebase por su gran versatilidad y fácil integración en aplicaciones móviles, para el almacenamiento se utilizó su base de datos firestore, una base de datos no relacional donde se almacena la información de los usuarios y tutorías.

Manejo de versiones y control del código:

Para el control de versiones y almacenamiento del código, se utilizó GitHub, una plataforma que permite gestionar cambios, trabajar de manera colaborativa y garantizar la integridad del proyecto.

5.2 Requisitos del hardware

Para garantizar un desempeño adecuado de la aplicación y una experiencia fluida para los usuarios, se definen los siguientes requisitos de hardware, tanto para dispositivos Android como IOS.

Requisitos para sistema operativo Android

Sistema operativo: El dispositivo debe tener Android 6.0 o mayor.

Memoria RAM: Para el correcto funcionamiento el dispositivo debe tener un mínimo de 2GB.

Memoria interna: El dispositivo deberá contar con mínimo 8 GB de espacio y 100 MB como mínimo para la instalación de la aplicación.

Conectividad: El dispositivo deberá tener conexión a internet o datos móviles para poder usar la aplicación.

Requisitos para sistema operativo IOS

Memoria RAM: Para el correcto funcionamiento el dispositivo debe tener un mínimo de 2GB.

Memoria interna: El dispositivo deberá contar con mínimo 8 GB de espacio y 100 MB como mínimo para la instalación de la aplicación.

Conectividad: El dispositivo deberá tener conexión a internet o datos móviles para poder usar la aplicación.

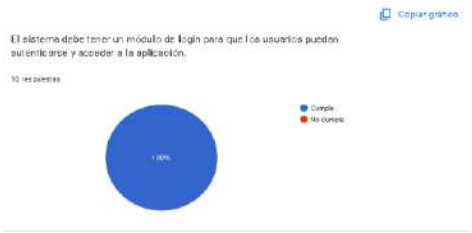
6 Pruebas del software

6.1 Inspección de software (Validación y Verificación) (RQF-RQNF)

En esta sección se realizan las actividades de validación y verificación del software para asegurar que la aplicación cumple con los requisitos funcionales y no funcionales definidos. La validación busca confirmar que el producto final satisface las necesidades de los usuarios (estudiantes y docentes) en cuanto a usabilidad, eficiencia y funcionalidades, como el agendamiento de tutorías y la mensajería en tiempo real.

Figura 37

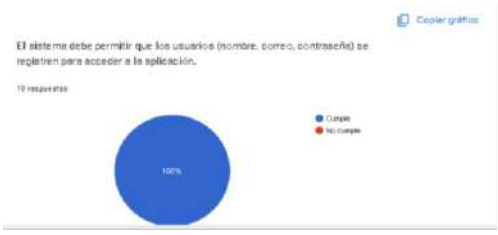
Inspección de Software



Nota: Autoría propia (2024)

Figura 38

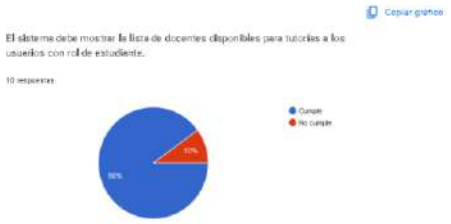
Inspección de Software



Nota: Autoría propia (2024)

Figura 39

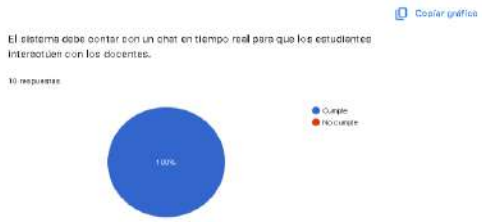
Inspección de Software



Nota: Autoría propia (2024)

Figura 40

Inspección de Software



Nota: Autoría propia (2024)

Figura 41

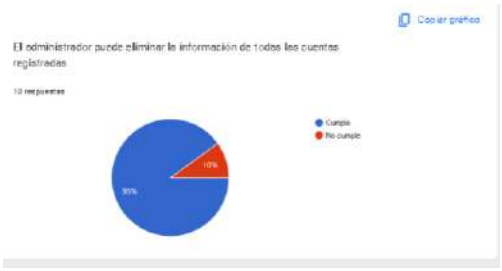
Inspección de Software



Nota: Autoría propia (2024)

Figura 42

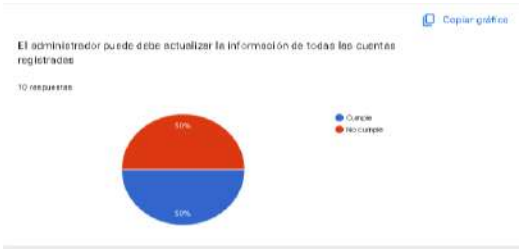
Inspección de Software



Nota: Autoría propia (2024)

Figura 43

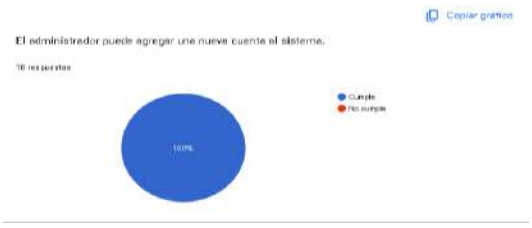
Inspección de Software



Nota: Autoría propia (2024)

Figura 44

Inspección de Software



Nota: Autoría propia (2024)

Figura 45

Inspección de Software



Nota: Autoría propia (2024)

Figura 46

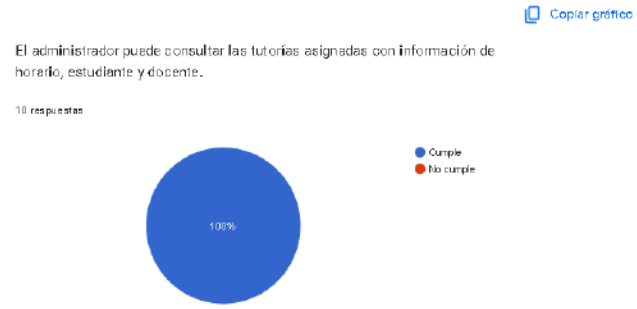
Inspección de Software



Nota: Autoría propia (2024)

Figura 47

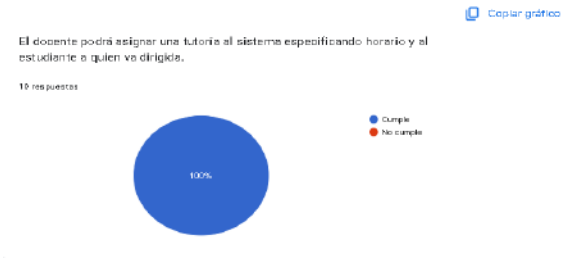
Inspección de Software



Nota: Autoría propia (2024)

Figura 48

Inspección de Software



Nota: Autoría propia (2024)

Figura 49

Inspección de Software



Nota: Autoría propia (2024)

Figura 50

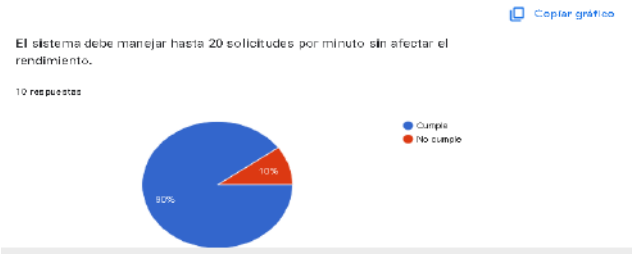
Inspección de Software



Nota: Autoría propia (2024)

Figura 51

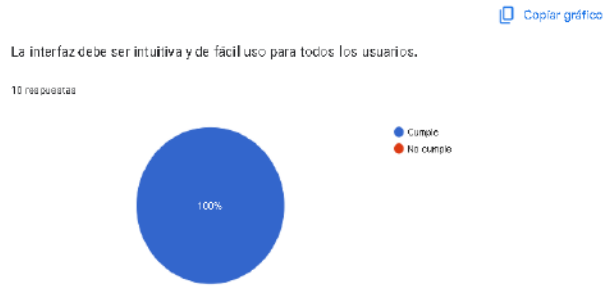
Inspección de Software



Nota: Autoría propia (2024)

Figura 52

Inspección de Software



Nota: Autoría propia (2024)

Figura 53

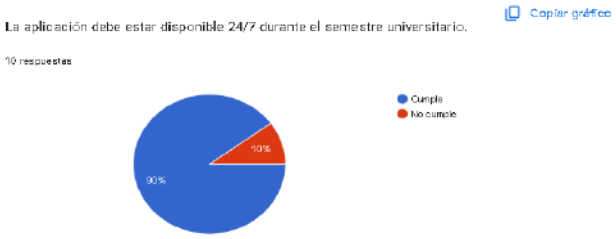
Inspección de Software



Nota: Autoría propia (2024)

Figura 54

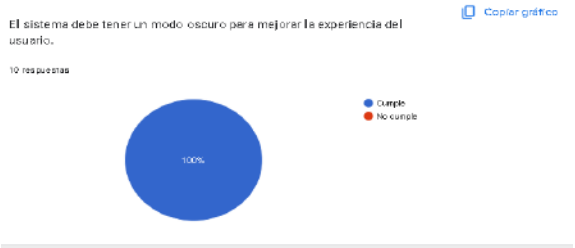
Inspección de Software



Nota: Autoría propia (2024)

Figura 55

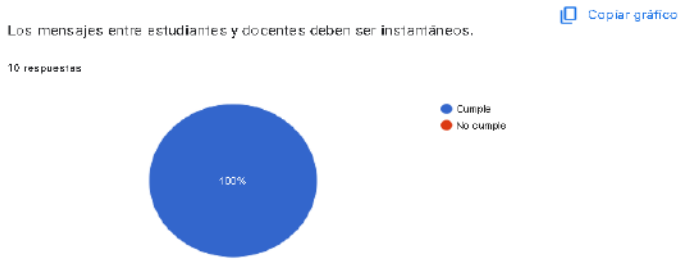
Inspección de Software



Nota: Autoría propia (2024)

Figura 56

Inspección de Software



Nota: Autoría propia (2024)