

EduConnect

**Plataforma para tutorías personalizadas y apoyo académico
para estudiantes de primaria y secundaria**

Autor: Alejandro Moya Ruiz Tutor: Patricio Fernández Flórez

Desarrollador de Aplicaciones Multiplataforma

Trabajo de fin de grado

Contenido

- 1. *Introducción* 1**
 - a. **Contextualización 1**
 - b. **Motivación 1**
 - c. **Objetivos 1**
 - d. **Estado del arte 2**
 - e. **Glosario 3**
 - i. Acrónimos3
 - ii. Definiciones.....3
- 2. *Planificación*..... 4**
 - a. **Metodología 4**
 - b. **Planificación del proyecto 5**
- 3. *Descripción del proyecto*..... 6**
 - a. **Requisitos de alto nivel 6**
 - b. **Tecnologías usadas y justificación de las tecnologías 6**
 - c. **Usuarios finales..... 7**
 - d. **Evaluación de riesgos 7**
 - e. **Reducción de riesgos 8**
- 4. *Presupuestos*..... 9**
 - a. **Costes directos 9**
 - b. **Costes indirectos 10**
 - c. **Otros costes..... 10**
 - d. **Costes totales 11**
- 5. *Análisis de requisitos* 11**
 - a. **Catálogo de actores..... 11**

b.	Requisitos funcionales	11
c.	Requisitos no funcionales	13
d.	Diagrama de casos de uso	14
6.	<i>Diseño</i>	15
a.	Modelo conceptual de datos	15
b.	Diagrama de interacción	16
c.	Diseño lógico de la arquitectura	17
d.	Diseño físico	17
7.	<i>Implementación del sistema</i>	17
a.	Estructura de la aplicación	18
8.	<i>Pruebas del sistema</i>	20
a.	Pruebas unitarias	20
b.	Pruebas de sistema	21
9.	<i>Conclusiones</i>	22
a.	Lecciones aprendidas	22
b.	Objetivos cumplidos	22
c.	Trabajo futuro	22
10.	<i>Referencias Bibliográficas</i>	23
11.	<i>Anexos</i>	23
a.	Manual de uso	23
b.	Porciones de código explicado	25

Índice de Figuras

Figura 1: Interfaz de Tusclasesparticulares	2
Figura 2 Imagen de Bel en Play Store	3
Figura 3 Diagrama de Gantt	5
Figura 4 Diagrama de casos de uso Estudiante	14
Figura 5 Diagrama de casos de uso Profesor	15
Figura 6 Modelo conceptual de datos	15
Figura 7 Diagrama de crear evento en calendario	16
Figura 8 Diagrama de consulta de profesores	16
Figura 9 Diseño lógico	17
Figura 10 Diseño físico	17
Figura 11 Controlador	18
Figura 12 Vista	18
Figura 13 Modelo	18
Figura 14 Rutas	19
Figura 15 BD	19
Figura 16 Landing page	23
Figura 17 Login	24
Figura 18 Registro de usuarios	24
Figura 19 Página de inicio	24
Figura 20 Añadir evento	25
Figura 21 Filtro profesores	25
Figura 22 Todos los usuarios	25

1. Introducción

a. Contextualización

Durante los últimos cuatro años, la formación ha experimentado un cambio significativo impulsado por la digitalización, especialmente durante la pandemia. Hoy en día, el 70 % de los jóvenes adultos eligen plataformas de formación en línea para su desarrollo profesional o académico. La pandemia aceleró la adopción de tecnologías existentes, lo que generó un aumento en la oferta académica en línea. En España, el 74 % de los alumnos activos en formación participan en programas a distancia, principalmente entre los 18 y 24 años.

b. Motivación

Teniendo en cuenta la situación mencionada anteriormente, la creación de una aplicación que proporcione tutorías adicionales se convierte en un recurso vital para garantizar que todos los estudiantes, sin importar sus circunstancias personales o habilidades individuales, tengan acceso equitativo a recursos educativos de alta calidad. Esta aplicación no solo estaría dirigida a aquellos que encuentran dificultades con ciertos conceptos, sino que también serviría como una herramienta integral para mejorar la experiencia educativa en su totalidad. Al proporcionar una plataforma centralizada que ofrezca tutorías, materiales de aprendizaje y evaluaciones, esta aplicación podría mitigar las barreras geográficas y socioeconómicas que a menudo impiden el acceso a una educación de calidad. Además, al ampliar su alcance y funcionalidad, la aplicación podría adaptarse a las necesidades específicas de cada estudiante, brindando un apoyo personalizado que fomente un aprendizaje más efectivo y significativo. En última instancia, la implementación de esta aplicación no solo podría mejorar el desempeño académico de los estudiantes, sino que también promovería la inclusión y la equidad en el ámbito de la educación a distancia, contribuyendo así a la construcción de una sociedad más igualitaria y educada.

c. Objetivos

- Desarrollar una aplicación que permita a los profesores ponerse en contacto con los usuarios que soliciten sus servicios.
- Desarrollar una aplicación que permita a los alumnos que están registrados contactar con los profesores y compartir apuntes y tareas con ellos.
- Desarrollar una aplicación que contenga calendario, recordatorios y que permita hacer tutorías.

d. Estado del arte

Hay una gran cantidad de plataformas y aplicaciones que prestan estos servicios o que son muy similares, estas son las más populares:

- Tusclasesparticulares: Permite a los estudiantes buscar y contactar a profesores particulares para recibir tutorías en una variedad de materias y niveles educativos.



Figura 1: Interfaz de Tusclasesparticulares

- MiProfe: Ofrece tutorías en línea con profesores cualificados en una amplia gama de materias, incluyendo matemáticas, ciencias, idiomas y más.

- Bel · Clases particulares: Ofrecen clases particulares, apoyo escolar y talleres grupales para niños, adolescentes y estudiantes, a través de una comunidad confiable de profesores particulares y tutores escolares.



Figura 2 Imagen de Bel en Play Store

e. Glosario

i. Acrónimos

ORM: Un ORM (Object-Relational Mapping) es una técnica de programación que permite convertir datos entre sistemas incompatibles utilizando lenguajes de programación orientados a objetos. Básicamente, un ORM facilita la interacción con bases de datos relacionales, permitiendo manipular datos de la base de datos utilizando objetos del lenguaje de programación, sin necesidad de escribir SQL directamente. Esto simplifica el código y mejora la productividad del desarrollo al manejar automáticamente las tareas de CRUD (crear, leer, actualizar, eliminar).

ii. Definiciones

Landing page: Una landing page, o página de aterrizaje, es una página web diseñada específicamente para convertir visitantes en leads o clientes. Su objetivo principal es enfocar la atención del usuario en una oferta concreta y guiarlo a realizar una acción específica, como completar un formulario, suscribirse a un boletín, descargar un recurso o realizar una compra.

2. Planificación

a. Metodología

Las metodologías de desarrollo de software ofrecen diferentes enfoques para gestionar el proceso de creación de programas informáticos. El Modelo en Cascada sigue una secuencia lineal de fases, desde la concepción hasta la implementación, mientras que el Prototipado involucra la creación de versiones preliminares del software para obtener retroalimentación temprana. Por otro lado, el Modelo Incremental desarrolla el software en etapas sucesivas y entregas incrementales, mientras que las metodologías ágiles como Scrum y Kanban priorizan la colaboración, la adaptación al cambio y la entrega continua de software funcional. Además, el enfoque DevOps integra las prácticas de desarrollo y operaciones para mejorar la velocidad y la calidad en el despliegue de software. Estas metodologías ofrecen distintas herramientas y procesos para abordar las necesidades y desafíos específicos del desarrollo de software, permitiendo a las organizaciones elegir el enfoque más adecuado según sus requisitos y objetivos.

La metodología de desarrollo que voy a utilizar para este proyecto será el modelo en cascada, porque se ajusta mejor a la naturaleza del proyecto, que tiene requisitos bien definidos desde el principio y no se anticipan grandes cambios en el alcance o los objetivos. Con el modelo en cascada, podemos seguir un enfoque secuencial y estructurado, completando cada fase, como la definición de requisitos, el diseño, la implementación, las pruebas y el despliegue, antes de pasar a la siguiente. Esto proporcionará una clara delimitación de las etapas del proyecto y permitirá una planificación precisa de los recursos y los tiempos. Además, al tener una visión clara de los requisitos desde el principio, podemos minimizar los riesgos asociados con cambios de última hora y asegurar una entrega oportuna del producto final.

b. Planificación del proyecto

Detallo la planificación en este diagrama, donde se pueden observar los tres grandes incrementos de los que consta este proyecto:

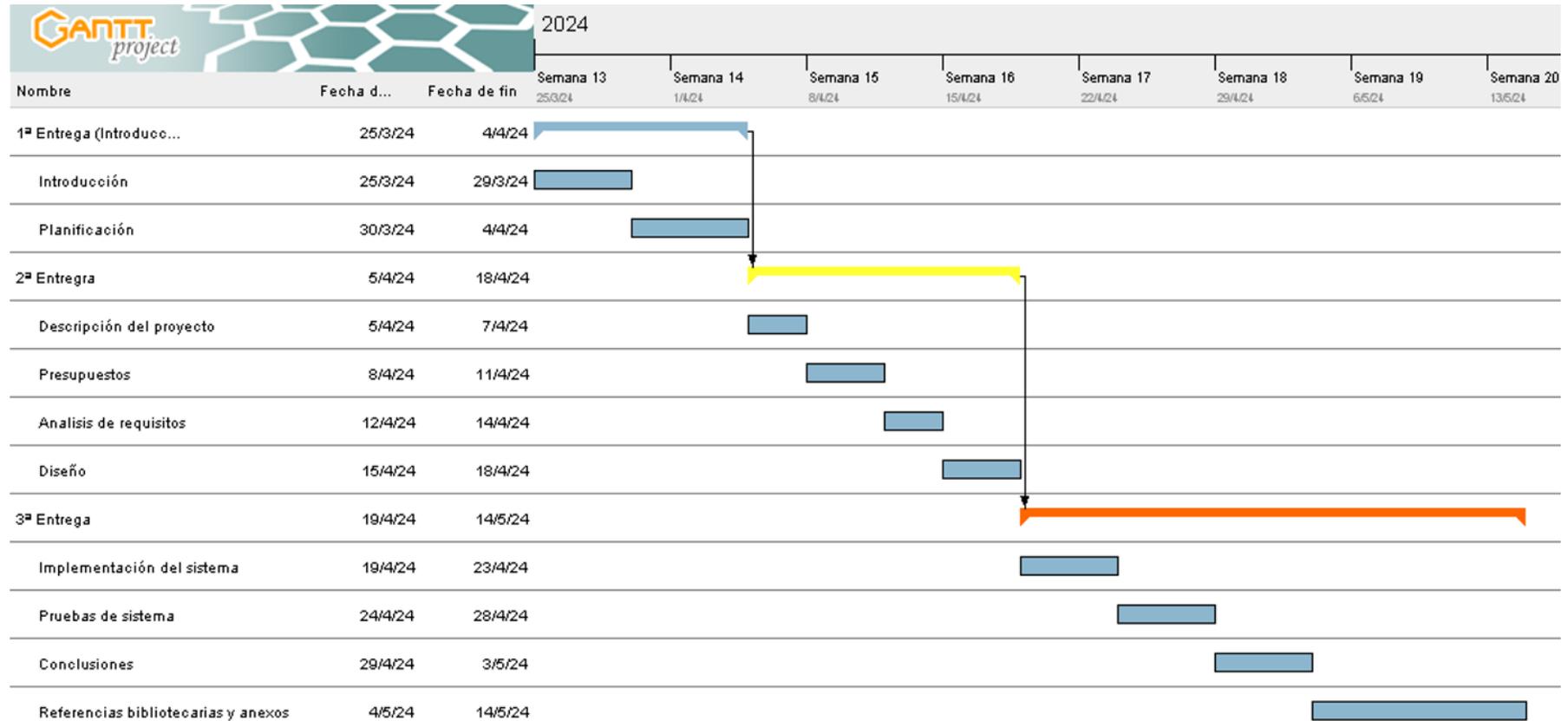


Figura 3 Diagrama de Gantt

3. Descripción del proyecto

a. Requisitos de alto nivel

- REQ1. Registro de cuenta para estudiantes y profesores: Permitir que ambos grupos creen cuentas con información básica.
- REQ2. Exploración de tutorías por materia y disponibilidad: Interfaz para buscar tutorías disponibles según la materia y horario.
- REQ3. Programación de sesiones de tutoría: Función para que los estudiantes reserven sesiones según la disponibilidad de los profesores.
- REQ4. Mensajería instantánea entre estudiantes y profesores: Sistema de mensajería integrado para comunicación directa.
- REQ5. Calendario y recordatorios de sesiones: Mostrar calendario con sesiones programadas y enviar recordatorios automáticos.
- REQ6. Seguimiento del progreso del estudiante: Herramientas para que los estudiantes monitoreen su rendimiento académico.
- REQ7. Gestión de sesiones de tutoría por parte de los profesores: Visualización y administración de las sesiones programadas.
- REQ8. Calificación y retroalimentación de las sesiones: Función para que los profesores evalúen y den feedback a los estudiantes.
- REQ9. Compartir recursos educativos: Capacidad para que los profesores compartan materiales de estudio.
- REQ10. Integración de videoconferencias: Implementación de herramienta de videoconferencia para sesiones virtuales en tiempo real.

b. Tecnologías usadas y justificación de las tecnologías

En la construcción de esta aplicación web, se ha optado por utilizar Laravel como nuestro principal framework de desarrollo debido a sus numerosas ventajas. Laravel, con su arquitectura basada en el patrón MVC (Modelo-Vista-Controlador), facilita la separación de lógica de negocios y presentación, permitiendo un desarrollo más organizado y mantenible. Además, la herramienta de línea de comandos Artisan proporciona una potente automatización de tareas, que simplifica y acelera el proceso de desarrollo. Además de Laravel, estamos integrando varias tecnologías adicionales para enriquecer las funcionalidades y mejorar la experiencia del usuario de nuestra aplicación. Entre estas tecnologías se incluyen:

- Bootstrap: Este framework de CSS es fundamental para el diseño de una interfaz de usuario responsiva y atractiva. Bootstrap nos permite utilizar una amplia gama de componentes de diseño y estilos predefinidos, garantizando una presentación coherente y profesional en todos los dispositivos.
- HTML y JavaScript: Para la estructura y comportamiento dinámico de la aplicación, utilizamos HTML y JavaScript. HTML es la columna vertebral del contenido de la web, mientras que JavaScript añade interactividad y funcionalidad, permitiendo una experiencia de usuario más dinámica e intuitiva.
- FullCalendar API: Para la gestión de eventos y calendarios, integramos la API de FullCalendar[2]. Esta herramienta proporciona una interfaz interactiva y personalizable para visualizar y gestionar eventos, lo cual es esencial para funcionalidades como la programación de citas o la planificación de actividades.
- Pusher: Para las funcionalidades en tiempo real, como las notificaciones y la actualización en vivo de datos, implementamos Pusher. Esta tecnología permite una comunicación bidireccional en tiempo real entre el cliente y el servidor, mejorando la interactividad y la capacidad de respuesta de la aplicación.

En resumen, la combinación de Laravel con Bootstrap, HTML, JavaScript, FullCalendar y Pusher [3] está permitiendo desarrollar una aplicación completa, funcional y segura, capaz de satisfacer las necesidades de nuestros usuarios de manera eficiente y efectiva.

c. Usuarios finales

La aplicación está dirigida a dos grupos principales de usuarios: estudiantes con dificultades académicas de diversos niveles educativos que necesitan apoyo en materias específicas y prefieren la enseñanza personalizada, y profesores dispuestos a ofrecer clases particulares a distancia, con experiencia en la enseñanza, flexibilidad horaria, interés en obtener ingresos adicionales y disfrute por la enseñanza individualizada.

d. Evaluación de riesgos

- R1.Privacidad y seguridad de los datos: Garantizar que la información personal y académica de los estudiantes esté protegida contra posibles violaciones de seguridad o filtraciones de datos.
- R2.Problemas técnicos: Abordar errores de programación, fallos en el rendimiento de la aplicación y compatibilidad con diferentes dispositivos y sistemas operativos para garantizar una experiencia de usuario fluida.

- R3. Baja adopción por parte de los usuarios: Asegurarse de que la aplicación sea bien recibida y utilizada por los estudiantes, considerando la interfaz de usuario, la facilidad de uso y la disponibilidad de características relevantes.
- R4. Competencia en el mercado: Diferenciar la aplicación ofreciendo características únicas y proporcionando valor agregado para destacar entre otras aplicaciones educativas similares disponibles para los estudiantes.
- R5. Calidad del contenido educativo: Asegurar que el contenido proporcionado en la aplicación sea de alta calidad y esté actualizado para garantizar su eficacia como herramienta de aprendizaje.
- R6. Cumplimiento normativo: Cumplir con las regulaciones y políticas educativas relevantes, obteniendo permisos o licencias específicas si es necesario.
- R7. Incumplimiento de plazos de desarrollo: Gestionar el cronograma de manera efectiva y comunicarse transparentemente con el equipo de desarrollo para evitar retrasos en el lanzamiento de la aplicación.

e. Reducción de riesgos

- RR1. Privacidad y seguridad de los datos:
- Implementar medidas de seguridad robustas y seguir las regulaciones de protección de datos.
 - Realizar auditorías de seguridad regulares para identificar y corregir posibles vulnerabilidades. [\(Mitiga a R1\)](#)
- RR2. Problemas técnicos:
- Realizar pruebas exhaustivas en diferentes dispositivos y sistemas operativos.
 - Mantener un equipo de desarrollo calificado y flexible para abordar problemas técnicos de manera eficaz. [\(Mitiga a R2\)](#)
- RR3. Baja adopción por parte de los usuarios:
- Realizar investigaciones de mercado para comprender las necesidades de los usuarios y mejorar la experiencia del usuario en consecuencia.
 - Implementar estrategias de marketing efectivas para promover la aplicación y aumentar su adopción. [\(Mitiga a R3\)](#)
- RR4. Competencia en el mercado:

- Diferenciar la aplicación ofreciendo características únicas y proporcionando un valor agregado claro.
- Mantenerse actualizado sobre las tendencias del mercado y realizar ajustes según sea necesario. ([Mitiga a R4](#))

RR5. Calidad del contenido educativo:

- Colaborar con expertos en educación para desarrollar contenido de alta calidad y relevante.
- Actualizar regularmente el contenido educativo para mantener su relevancia y precisión. ([Mitiga a R5](#))

RR6. Cumplimiento normativo:

- Investigar y cumplir con las regulaciones educativas relevantes.
- Obtener las licencias y permisos necesarios antes del lanzamiento de la aplicación. ([Mitiga a R6](#))

RR7. Incumplimiento de plazos de desarrollo:

- Establecer un plan de proyecto para evitar retrasos.
- Utilizar metodologías ágiles para adaptarse a los cambios y entregar incrementos de producto de manera oportuna. ([Mitiga a R7](#))

4. Presupuestos

El presupuesto abarca todos los gastos incurridos durante el proceso de desarrollo de la aplicación, desde los elementos más fundamentales como el equipo informático necesario para la programación inicial, hasta los recursos utilizados para llevar a cabo pruebas en dispositivos y asegurar la funcionalidad adecuada de la aplicación. En resumen, representa el total de todos los costos asociados con la creación y validación de la aplicación antes de su lanzamiento al mercado.

a. Costes directos

Considerando que mi tarifa por hora es de 30€ más IVA y que planeo trabajar 100 horas durante los 55 días de desarrollo del proyecto, el costo total de mi trabajo para el proyecto sería de 3000€ más IVA. Esto abarca todas las horas dedicadas a la programación, diseño y demás tareas relacionadas con el desarrollo de la aplicación. El presupuesto está detallado en la siguiente tabla:

Concepto	N.º de Horas
Análisis	15H
Implementación	65H
Pruebas	5H
Documentación	15H
Total	<i>100H</i>

b. Costes indirectos

Estos costes representan los gastos en los que incurre el trabajador para realizar las tareas asignadas. Esto incluye no solo el valor de su tiempo y esfuerzo, calculado en base a su tarifa por hora, sino también todos los recursos, tanto de software como de hardware, que se utilizaron en mayor o menor medida para llevar a cabo el proyecto.

Coste de software:

Software	Precio
Visual Studio Code	0€
Microsoft 365	100€
Gantt Project	0€
Total	100€

Costes de hardware:

La amortización del ordenador se calcula dividiendo su costo total de 1000€ por su durabilidad de 5 años, lo que resulta en un gasto anual de 200€. Dado que 5 años equivalen a 1826 días y cada día tiene un valor de 0.547€, el costo diario de uso del ordenador es de 0.547€. Trabajando 55 días con el ordenador, el gasto total de amortización durante este periodo sería de 30,120€.

Concepto	Total
Ordenador sobremesa	30,12€

c. Otros costes

Costes de luz e internet:

Concepto	Total (€)
Luz	65€
Internet	50€
Total	115€

d. Costes totales

Concepto	Total
Costes directos	3000€
Costes indirectos	130,12€
Otros costes	115€
Total	3.245,12€

5. Análisis de requisitos

a. Catálogo de actores

- **Profesores:** Profesionales cualificados en materias específicas, listos para ofrecer tutorías personalizadas a los alumnos.
- **Estudiantes:** Los propios alumnos que buscan mejorar su rendimiento académico en diferentes materias
- **Administradores:** Encargados de gestionar la plataforma, asignar tutores, y resolver cualquier problema técnico o administrativo.

b. Requisitos funcionales

- RF1. Registro de cuenta para estudiantes y profesores (Satisface [REQ1]):
- La plataforma permitirá a usuarios registrarse como estudiantes o profesores.
 - Se requerirá información básica como nombre, correo electrónico y contraseña.
 - Se proporcionará un proceso de verificación para garantizar la autenticidad de las cuentas.
- RF2. Exploración de tutorías por materia y disponibilidad (Satisface [REQ2]):
- Se desarrollará una interfaz de búsqueda que permita a los usuarios encontrar tutorías según la materia deseada.

- b. Los usuarios podrán filtrar las tutorías disponibles por horario y ubicación si es relevante.
- c. Se mostrarán resultados detallados que incluyan la descripción de la tutoría y la información del profesor.

RF3. Programación de sesiones de tutoría (Satisface [REQ3]):

- a. Los estudiantes podrán reservar sesiones de tutoría según la disponibilidad proporcionada por los profesores.
- b. Se permitirá a los estudiantes cancelar o reprogramar sesiones con un aviso previo determinado.

RF4. Mensajería instantánea entre estudiantes y profesores (Satisface [REQ4]):

- a. Se integrará un sistema de mensajería en la plataforma para facilitar la comunicación directa entre estudiantes y profesores.
- b. Los usuarios podrán intercambiar mensajes de texto, archivos adjuntos y enlaces relacionados con la tutoría.

RF5. Calendario y recordatorios de sesiones (Satisface [REQ5]):

- a. Se mostrará un calendario que refleje las sesiones de tutoría programadas por el usuario.
- b. Se enviarán recordatorios automáticos a los usuarios antes de cada sesión programada.

RF6. Seguimiento del progreso del estudiante (Satisface [REQ6]):

- a. Se proporcionarán herramientas para que los estudiantes puedan monitorear su rendimiento académico, como estadísticas de progreso y resultados de pruebas.

RF7. Gestión de sesiones de tutoría por parte de los profesores (Satisface [REQ7]):

- a. Los profesores podrán visualizar y administrar las sesiones de tutoría programadas con sus estudiantes.
- b. Se permitirá a los profesores aceptar, rechazar o modificar sesiones según su disponibilidad.

- RF8. Calificación y retroalimentación de las sesiones (Satisface [REQ8]):
- a. Los profesores podrán evaluar el desempeño de los estudiantes durante las sesiones de tutoría y proporcionar retroalimentación.
 - b. Los estudiantes tendrán acceso a las calificaciones y comentarios para mejorar su aprendizaje.
- RF9. Compartir recursos educativos (Satisface [REQ9]):
- a. Se habilitará una función para que los profesores compartan materiales educativos relevantes, como documentos, presentaciones o enlaces a recursos en línea.
- RF10. Integración de videoconferencias (Satisface [REQ10]):
- a. Se implementará una herramienta de videoconferencia para permitir sesiones de tutoría virtuales en tiempo real entre estudiantes y profesores.

c. Requisitos no funcionales

Los requisitos no funcionales son todas las características que el sistema debe cumplir aparte de la funcionalidad que realiza. Los criterios más estandarizados son los criterios de la ISO/IEC 25010 [1] por lo tanto, son los que se van a seguir:

1. Rendimiento: Definimos rendimiento como la capacidad de desempeño en torno a los recursos de los que disponemos. Se mantendrá la web lo más ligera posible para que los recursos del ordenador se centren en cargar los datos de la base de datos.
2. Portabilidad: Definimos portabilidad como la capacidad de adaptarse a otros sistemas. Dado que es una aplicación web se ejecuta desde un explorador y será accesible desde cualquier dispositivo que disponga de un navegador.
3. Compatibilidad: la aplicación debe ser compatible con diferentes dispositivos y navegadores web. Dado que es una aplicación web se ejecuta desde un navegador y será accesible desde cualquier dispositivo que disponga de un explorador.
4. Usabilidad: la aplicación debe ser fácil de usar y navegar, con una interfaz intuitiva y clara. Los formularios serán lo más sencillos posibles, de esta manera será intuitivo y rápido.
5. Fiabilidad: Fiabilidad es la capacidad de confianza que se tiene en las operaciones que se realizan en la aplicación. EduConnect tratará los datos personales y financieros de los

usuarios mediante medidas de seguridad sólidas, como la encriptación y políticas de privacidad claras.

6. Seguridad: la aplicación debe ser segura y proteger la información personal de los usuarios y los datos de actividad física. EduConnect garantizará seguridad mediante la implementación de encriptación de datos, autenticación segura para el acceso de usuarios, políticas de privacidad transparentes, medidas de protección contra ataques cibernéticos, actualizaciones regulares para abordar vulnerabilidades y auditorías periódicas de seguridad.
7. Mantenibilidad: la aplicación debe ser fácil de mantener y actualizar para garantizar su correcto funcionamiento a lo largo del tiempo. La aplicación se diseñará de forma modular para simplificar la corrección de errores y la incorporación de nuevas funciones. El código estará limpio, bien documentado y se someterá a pruebas automatizadas para garantizar su funcionalidad.

d. Diagrama de casos de uso

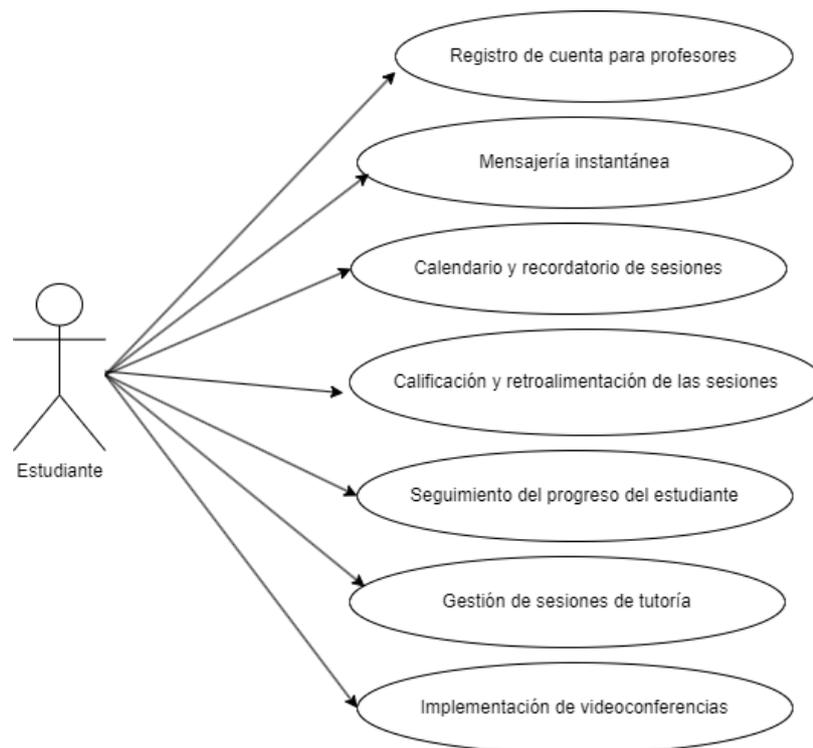


Figura 4 Diagrama de casos de uso Estudiante

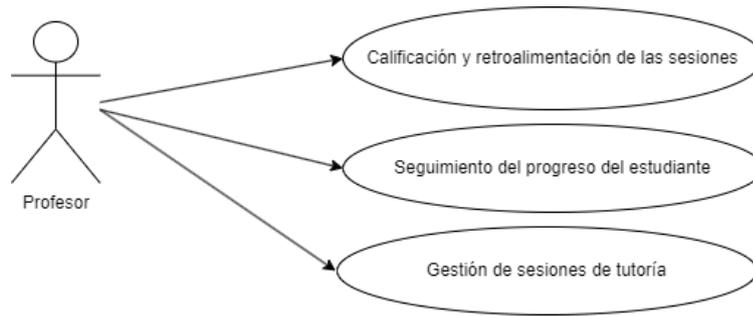


Figura 5 Diagrama de casos de uso Profesor

6. Diseño

a. Modelo conceptual de datos

En el modelo conceptual de datos se describe la información que se almacenará y cómo se relacionan los diferentes registros.

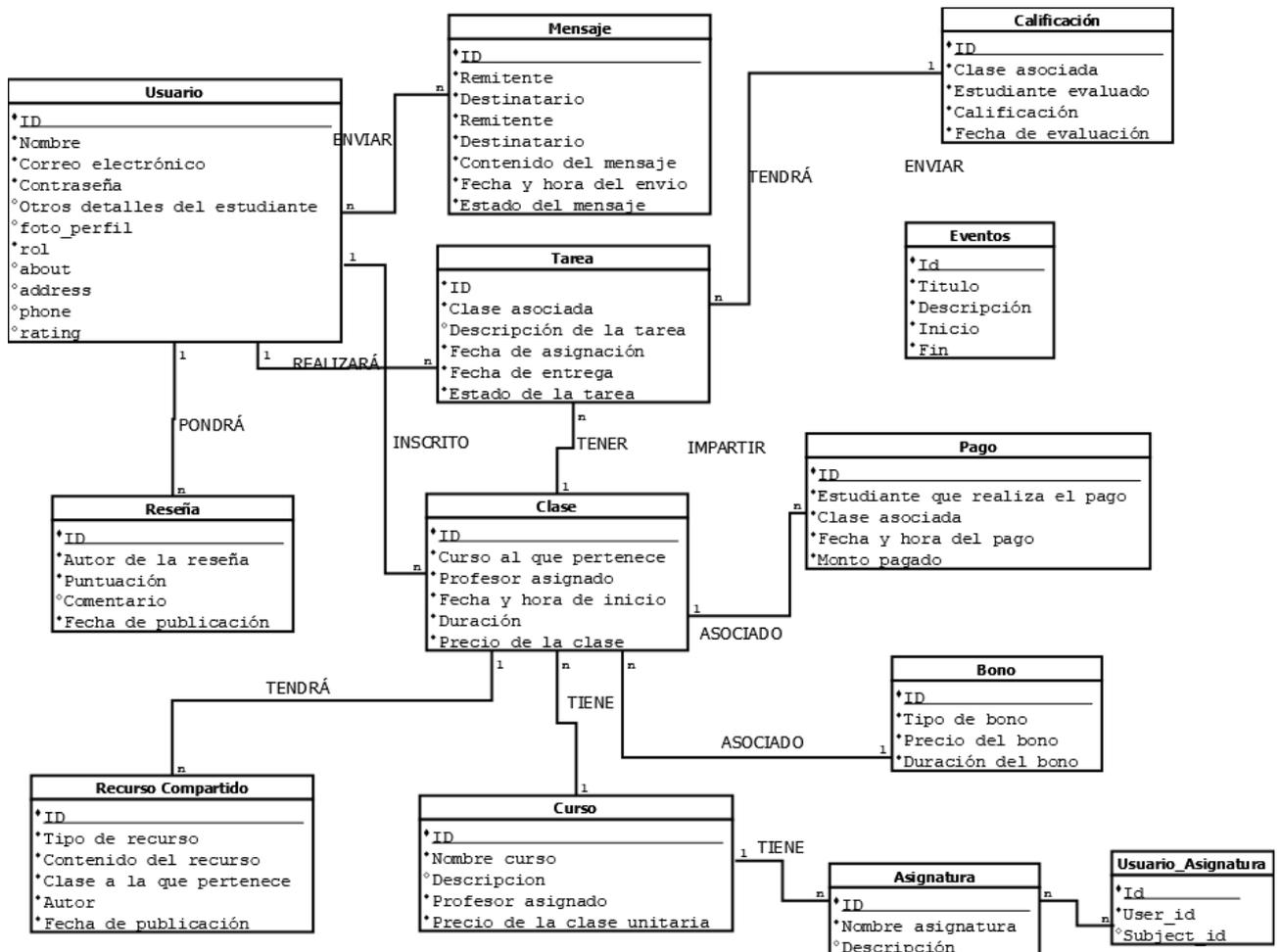


Figura 6 Modelo conceptual de datos

b. Diagrama de interacción

Un diagrama de interacción es una herramienta en UML que ilustra cómo los objetos interactúan entre sí dentro de un sistema. Se enfoca en mostrar los mensajes que los objetos se envían y reciben, ayudando a visualizar cómo se comunican y colaboran para lograr ciertas funcionalidades.



Figura 7 Diagrama de crear evento en calendario

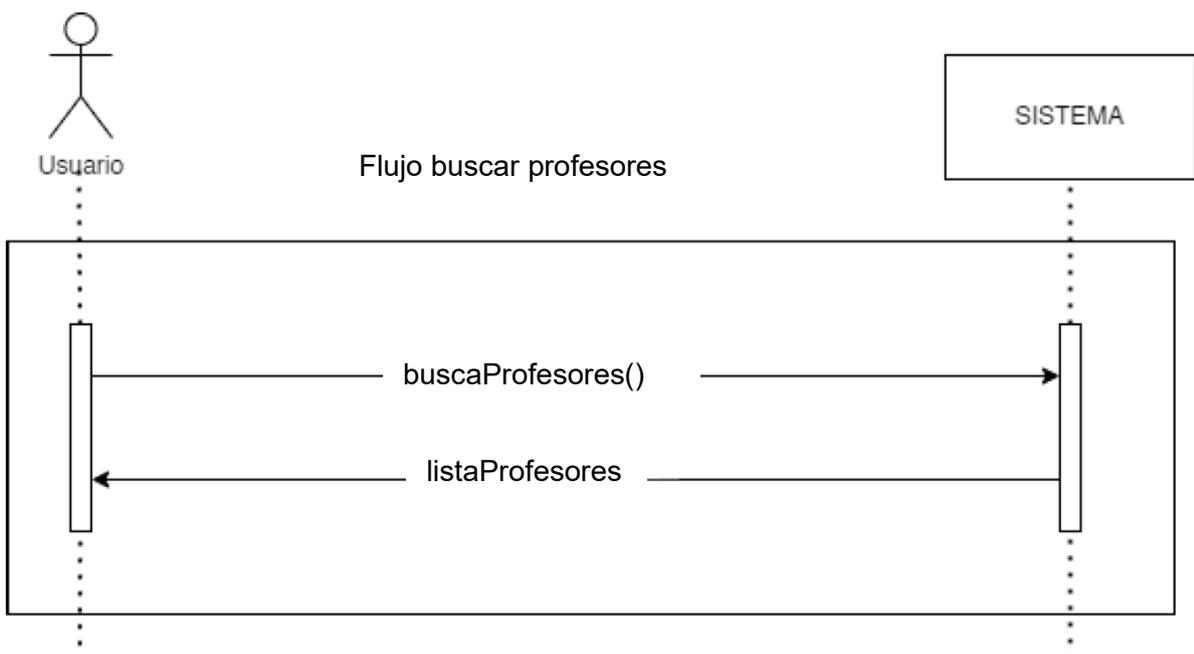


Figura 8 Diagrama de consulta de profesores

c. Diseño lógico de la arquitectura

Los diagramas de diseño lógico de la arquitectura proporcionan una representación visual que nos ayuda a entender cómo opera el sistema, mostrando sus componentes y cómo se relacionan entre sí.

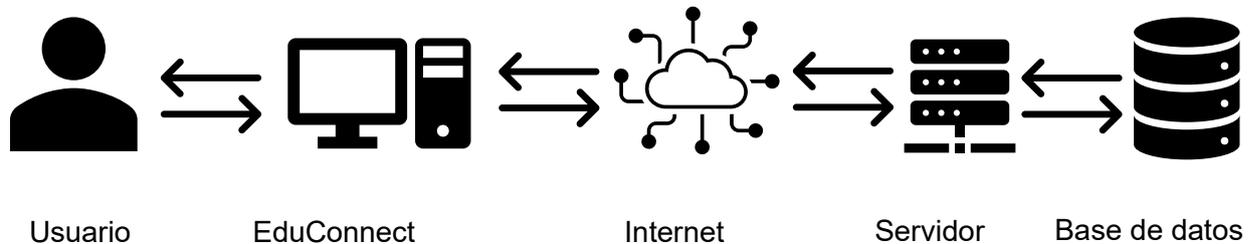


Figura 9 Diseño lógico

d. Diseño físico

El diseño físico de una aplicación se trata de planificar cómo se llevará a cabo la construcción y organización de esta, lo cual abarca decisiones como qué lenguajes de programación usar, qué plataforma de desarrollo escoger, qué base de datos emplear y cómo configurar los servidores. En nuestro caso, contamos con una infraestructura básica que consiste en una sola máquina actuando como servidor para ejecutar el servicio. Al ser una aplicación web, esto significa que es fácilmente accesible desde cualquier dispositivo móvil que disponga de un navegador web.

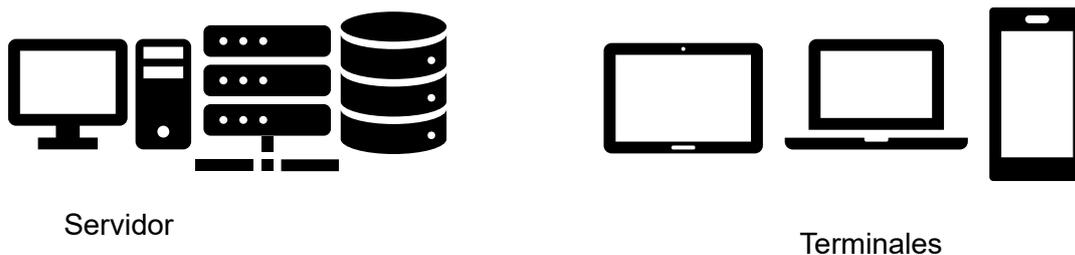


Figura 10 Diseño físico

7. Implementación del sistema

En el desarrollo de mi aplicación, he utilizado Laravel como el marco de trabajo principal debido a su robustez y flexibilidad. Laravel ha facilitado la creación de una arquitectura sólida y escalable, permitiéndome implementar funcionalidades de manera eficiente. Para la gestión de la base de datos, he aprovechado Eloquent ORM de Laravel, lo que ha simplificado considerablemente las operaciones CRUD y ha mejorado la interacción con la base de datos. Además, he utilizado Blade, el motor de plantillas de Laravel, para crear vistas dinámicas y responsivas, asegurando una

experiencia de usuario fluida. Por último, para la autenticación y autorización, he empleado el sistema de autenticación integrado de Laravel, garantizando la seguridad de los datos y la gestión de usuarios de manera eficaz.

a. Estructura de la aplicación

La estructura de la aplicación está organizada siguiendo el modelo MVC (Modelo-Vista-Controlador) proporcionado por Laravel. El directorio de modelos contiene las representaciones de datos y las relaciones entre ellos, mientras que los controladores gestionan la lógica de la aplicación y las solicitudes del usuario. Las vistas, ubicadas en el directorio de recursos, se encargan de la presentación de la interfaz de usuario mediante el uso de Blade. Además, se han implementado rutas claras y específicas en el archivo de rutas para manejar la navegación y las operaciones de la aplicación de manera eficiente.

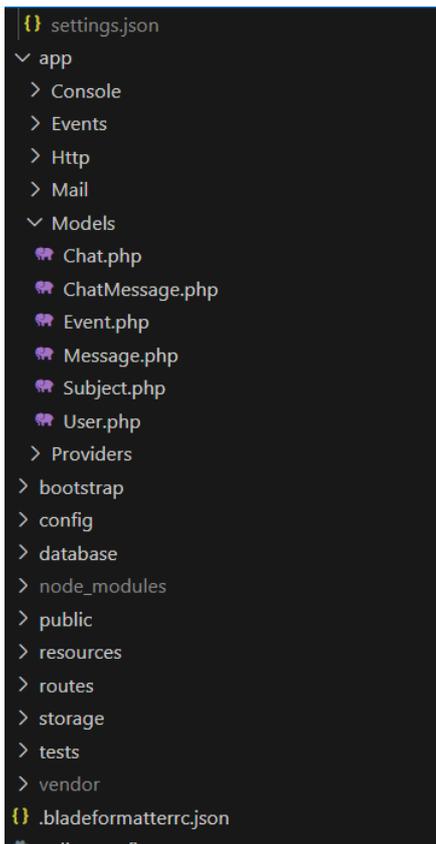


Figura 13 Modelo

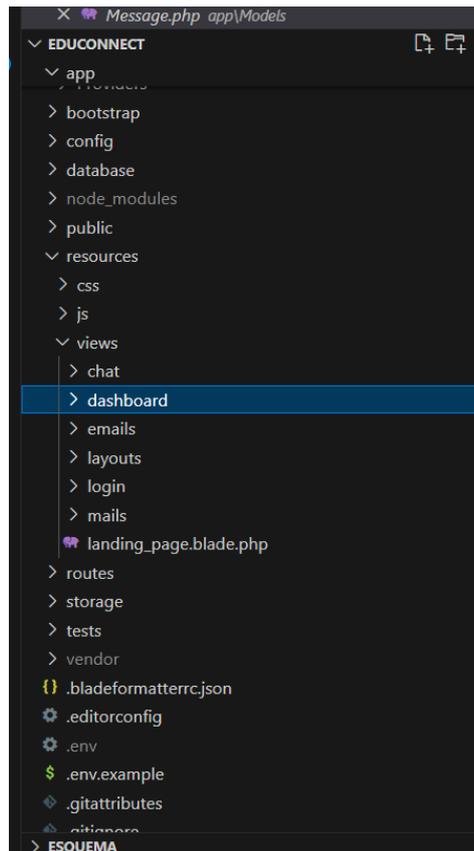


Figura 12 Vista

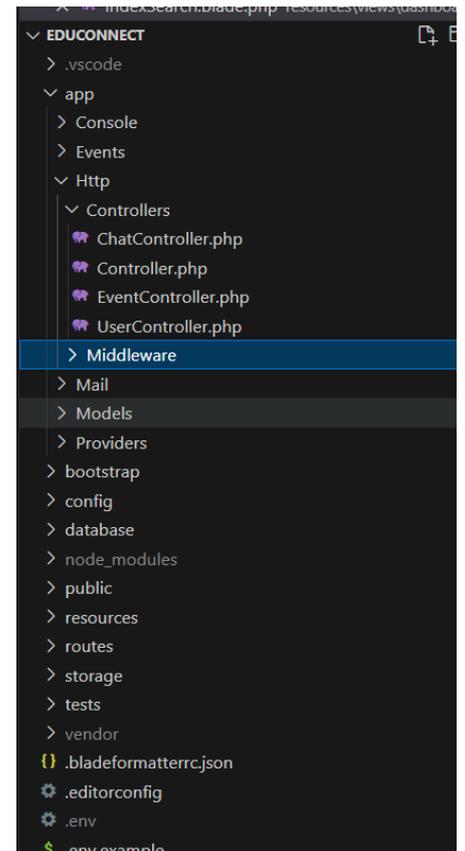


Figura 11 Controlador

```

21
22
23 Route::post('/register', [UserController::class, 'registerUser'])->name('register');
24
25 //***** */
26
27 Route::get('/recuperar', [UserController::class, 'showrecuperar'])->name('recuperar');
28
29 Route::get('/reset', [UserController::class, 'showreset'])->name('reset');
30
31
32 // Función que se ejecuta al enviar el formulario y que enviará el email al usuario
33 Route::post('/enviar-recuperar-contrasenia', [UserController::class, 'enviarRecuperar']);
34
35 // Formulario donde se modificará la contraseña
36 Route::get('/reiniciar-contrasenia/{token}', [UserController::class, 'formularioRecupe
37
38 // Función que actualiza la contraseña del usuario
39 Route::post('/actualizar-contrasenia', [UserController::class, 'actualizarContrasenia
40
41 //***** */
42
43
44 Route::post('/index', [UserController::class, 'login'])->name('custom-login');
45
46 Route::get('/home', [UserController::class, 'logados'])->name('home');
47
48 Route::get('/buscar_profesor', [UserController::class, 'searchTeachers'])->name('show
49
50 //*****CALENDARIO***** */
51 Route::get('/calendario', [EventController::class, 'index'])->name('calendar');
52
53 Route::put('/events/{eventId}', [EventController::class, 'update']);
54
55 Route::delete('/events/{eventId}', [EventController::class, 'delete']);

```

Figura 14 Rutas

La base de datos de mi aplicación en Laravel está estructurada en varias tablas, cada una cumpliendo funciones específicas. La tabla `users` gestiona la información de los usuarios registrados y cuenta con un campo `rol` que distingue entre estudiantes y profesores. La tabla `subjects` contiene los detalles de las materias, y la relación entre usuarios y materias se maneja en `user_subjects`. La tabla `events` registra todos los eventos del calendario, permitiendo una gestión eficiente de actividades y horarios. Para la funcionalidad de chat, se utilizan las tablas `chat` y `messages`. Las tablas `jobs`, `job_batches`, y `failed_jobs` gestionan las tareas en cola y sus estados. Además, `sessions` maneja las sesiones de usuario y `personal_access_tokens` los tokens de acceso personal. Finalmente, las tablas `cache` y `cache_locks` se utilizan para mejorar el rendimiento mediante almacenamiento en caché, y `migrations` controla las migraciones de la base de datos. La base de datos está alojada localmente en mi ordenador y es gestionada mediante el sistema de gestión de bases de datos MySQL.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a de
cache	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	
cache_locks	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	
chat	Examinar Estructura Buscar Insertar Vaciar Eliminar	70	InnoDB	utf8mb4_unicode_ci	16.0 KB	
events	Examinar Estructura Buscar Insertar Vaciar Eliminar	22	InnoDB	utf8mb4_unicode_ci	16.0 KB	
failed_jobs	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	
jobs	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	
job_batches	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	
messages	Examinar Estructura Buscar Insertar Vaciar Eliminar	70	InnoDB	utf8mb4_unicode_ci	32.0 KB	
migrations	Examinar Estructura Buscar Insertar Vaciar Eliminar	22	InnoDB	utf8mb4_unicode_ci	16.0 KB	
password_reset_tokens	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	
personal_access_tokens	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	
sessions	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8mb4_unicode_ci	48.0 KB	
subjects	Examinar Estructura Buscar Insertar Vaciar Eliminar	102	InnoDB	utf8mb4_unicode_ci	16.0 KB	
teachers	Examinar Estructura Buscar Insertar Vaciar Eliminar	17	InnoDB	utf8mb4_unicode_ci	32.0 KB	
teacher_subject	Examinar Estructura Buscar Insertar Vaciar Eliminar	15	InnoDB	utf8mb4_unicode_ci	48.0 KB	
users	Examinar Estructura Buscar Insertar Vaciar Eliminar	11	InnoDB	utf8mb4_unicode_ci	32.0 KB	
user_subject	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_unicode_ci	32.0 KB	
17 tabla(s)	Número de filas	336	InnoDB	utf8mb4_general_ci	464 KB	

Figura 15 BD

8. Pruebas del sistema

a. Pruebas unitarias

Prueba unitaria de creación, edición y eliminación de eventos en el calendario

Descripción: Esta prueba tiene como objetivo verificar si se cumple el requisito funcional 5 (rf5), que implica la capacidad de crear, editar y eliminar eventos en el calendario. Además de la creación del evento, se evaluará la capacidad de editar y eliminar eventos existentes para garantizar que la funcionalidad del calendario sea completa y versátil.

Pasos:

- Acceder al calendario en la plataforma.
- Seleccionar un día en el calendario para crear un nuevo evento.
- Completar los detalles del evento, como título, fecha, hora y descripción.
- Guardar el evento recién creado.
- Verificar que el evento se refleje correctamente en el calendario.
- Intentar editar el evento para modificar algún detalle, como la fecha o la descripción.
- Confirmar que los cambios se apliquen correctamente y se actualicen en el calendario.
- Intentar eliminar el evento.
- Verificar que el evento se elimine correctamente del calendario.

Resultado Esperado: Se espera que la plataforma permita a los usuarios crear eventos en el calendario, editar eventos existentes y eliminar eventos según sea necesario. Todos los cambios realizados deberían reflejarse correctamente en el calendario sin errores.

Resultado: La prueba de sistema ha sido exitosa. Se pudo crear un nuevo evento en el calendario sin errores, y este evento se reflejó correctamente en el calendario. Además, se pudo editar y eliminar el evento sin problemas, y todos los cambios se aplicaron correctamente en el calendario. No se registraron errores durante la ejecución de la prueba, y se verificó en la base de datos que el evento se creara, edite y elimine correctamente.

Prueba unitaria de exploración de tutorías según la materia deseada

Descripción: En esta prueba, se verificará si se satisface el requisito funcional 2 (rf2), que implica la capacidad de explorar tutorías según la materia deseada. Además de la búsqueda por materia, se evaluará si es posible visualizar la información detallada de las tutorías encontradas, como fecha y hora.

Pasos:

- Acceder a la sección de buscar nuevos profesores en la plataforma.
- Seleccionar la materia deseada de la lista de materias disponibles.
- Verificar que se muestren correctamente las tutorías relacionadas con la materia seleccionada.
- Comprobar si es posible visualizar información detallada de las tutorías, como fecha, hora y descripción.
- Realizar una búsqueda utilizando diferentes materias para asegurar la funcionalidad adecuada del sistema.

Resultado Esperado: Se espera que la plataforma permita a los usuarios explorar tutorías según la materia deseada, mostrando correctamente las tutorías disponibles y su información detallada. No se deberían detectar errores perceptibles durante la ejecución de la prueba.

Resultado: La prueba se completó exitosamente, cumpliendo con las expectativas establecidas. Se confirmó que el sistema permite explorar tutorías de acuerdo con la materia deseada sin errores perceptibles. Además, se verificó que la base de datos refleje correctamente las tutorías disponibles, y no se detectaron fallos en la consola durante la ejecución de la prueba.

b. Pruebas de sistema

Prueba de Sistema: Registro de Estudiantes o Profesores

Descripción: Esta prueba tiene como objetivo evaluar si se cumple el requisito de alto nivel 1 (REQ 1), que implica la capacidad de crear un usuario (estudiante o profesor) incluyendo todos los datos requeridos según las especificaciones del sistema.

Pasos:

- Acceder al formulario de registro en la plataforma.
- Seleccionar la opción de registro como estudiante o profesor.
- Completar todos los campos obligatorios del formulario, como nombre, dirección de correo electrónico, contraseña, etc.
- Enviar el formulario de registro.

Resultado Esperado: Se espera que la plataforma permita a los usuarios (estudiantes o profesores) registrarse satisfactoriamente, almacenando todos los datos proporcionados en la base de datos sin errores.

Resultado: La prueba de sistema ha sido exitosa. Después de la creación de un usuario nuevo, los datos se almacenan en la base de datos sin que se produzca ningún tipo de error, cumpliendo así con el requisito 1 (REQ 1).

9. Conclusiones

a. Lecciones aprendidas

Durante el desarrollo de este proyecto de clases particulares en línea, he adquirido conocimientos sobre el Modelo Vista Controlador (MVC), el manejo de rutas, y otras funcionalidades clave de Laravel. Además, he obtenido experiencia en el uso de PHP y HTML, lo que me ha permitido desarrollar y personalizar la lógica y la interfaz de usuario de manera efectiva. Estas habilidades combinadas han sido fundamentales para crear un sistema flexible y adaptable a las necesidades específicas del proyecto.

b. Objetivos cumplidos

En el desarrollo de mi proyecto, he cumplido con los siguientes objetivos:

Desarrollar una aplicación que permita a los profesores ponerse en contacto con los usuarios que soliciten sus servicios: He implementado un sistema de mensajería instantánea que facilita la comunicación directa entre profesores y estudiantes, permitiendo a los profesores responder a las solicitudes de tutoría de manera eficiente.

Desarrollar una aplicación que contenga calendario y recordatorios: He integrado un calendario en la plataforma que refleja las sesiones de tutoría programadas, y se envían recordatorios automáticos antes de cada sesión para asegurar la puntualidad y organización. Además, los estudiantes pueden reservar, cancelar y reprogramar sesiones de tutoría según la disponibilidad de los profesores.

Estos objetivos alcanzados han sido fundamentales para proporcionar una plataforma funcional y eficiente, que facilita la interacción entre estudiantes y profesores y mejora la gestión académica a través de herramientas prácticas y efectivas.

c. Trabajo futuro

Como trabajo futuro, tengo planeado abordar los siguientes objetivos:

- Integración de Videoconferencias: Se implementará una herramienta para permitir sesiones de tutoría virtual en tiempo real entre estudiantes y profesores, facilitando una interacción más directa y efectiva.
- Intercambio de Archivos y Enlaces: Se habilitará la funcionalidad para intercambiar archivos adjuntos y enlaces relacionados con la tutoría, mejorando el flujo de información entre estudiantes y profesores.

- **Monitoreo del Rendimiento Académico:** Se proporcionarán herramientas para que los estudiantes puedan monitorear su rendimiento académico, incluyendo estadísticas de progreso y resultados de pruebas.
- **Evaluación y Retroalimentación:** Los profesores podrán evaluar el desempeño de los estudiantes durante las sesiones de tutoría y proporcionar retroalimentación, permitiendo a los estudiantes mejorar basándose en la evaluación recibida.
- **Compartir Recursos Educativos:** Se habilitará una función para que los profesores compartan materiales educativos relevantes, enriqueciendo el proceso de aprendizaje.

Además, esta aplicación está pensada como un proyecto a largo plazo, con acceso como administrador y un sistema de pagos de alumnos a profesores. La aplicación es muy escalable, permitiendo la adición constante de nuevas funcionalidades. La principal razón de no haber cumplido todos los objetivos ha sido mayoritariamente el tiempo, ya que implementar todas estas funcionalidades requiere una considerable inversión de tiempo y recursos.

10. Referencias Bibliográficas

[1] ISO/IEC 25010 en: <https://iso25000.com/index.php/normas-iso-25000/iso-25010>

[2] [FullCalendar - JavaScript Event Calendar](#)

[3] [Pusher | Leader In Realtime Technologies](#)

11. Anexos

a. Manual de uso

Para poder acceder a EduConnect, primero debemos tener instalado PHP y Composer en nuestro ordenador. Luego, a través de la consola de comandos, arrancamos el servidor con el siguiente comando: `php artisan serve`. Con esto, ya tendremos acceso a la aplicación en localhost.



Figura 16 Landing page

Al acceder a localhost, primero veremos una landing page donde se describe de qué trata la aplicación. En la esquina superior derecha, encontraremos el acceso al login en un botón que pone "Únete a nosotros". Desde la página de login, veremos un enlace que dice "¿No tiene cuenta? Regístrese" el cual nos llevará al formulario de registro. Ahí, rellenamos los campos necesarios, asegurándonos de que el nombre de usuario tenga más de dos caracteres y la contraseña un mínimo de 8 caracteres. Una vez completado el registro, podemos iniciar sesión.

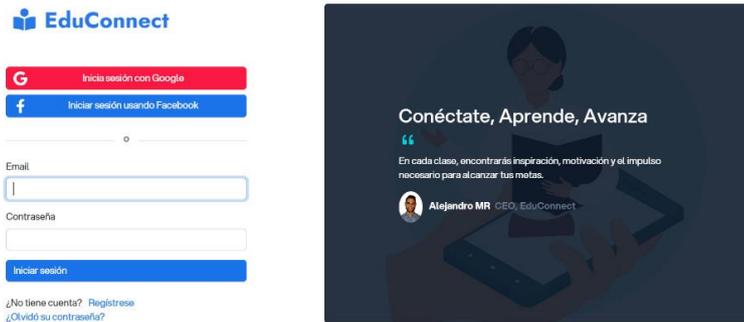


Figura 17 Login

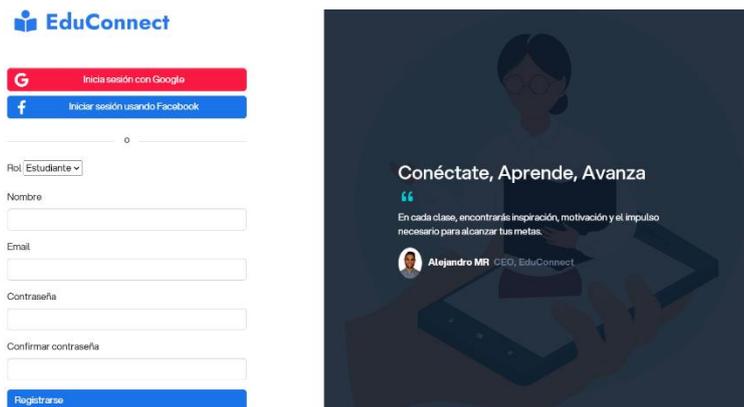


Figura 18 Registro de usuarios

Una vez que iniciemos sesión, accederemos a la página principal, donde podremos ver el panel de control. Desde este panel, tendremos la capacidad de navegar por todas las secciones de la aplicación. Para cerrar sesión, simplemente debemos hacer clic en "Cerrar sesión", ubicado en la esquina inferior izquierda.

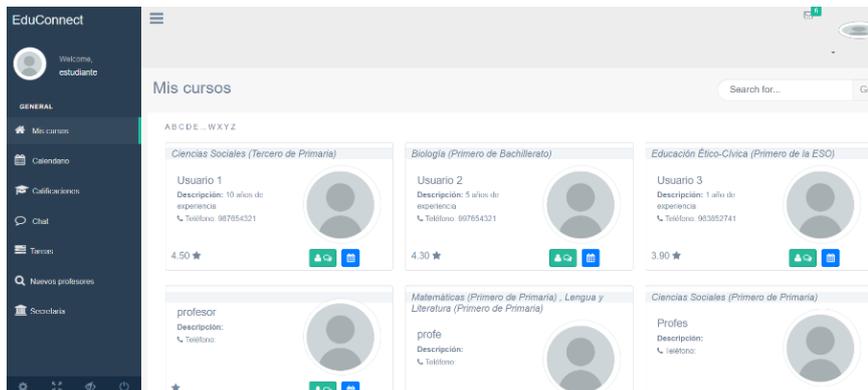


Figura 19 Página de inicio

b. Porciones de código explicado

```
// Manejador para el botón "Guardar" en el formulario de creación de eventos
$('#saveEventBtn').on('click', function() {
    // Obtener los datos del formulario
    var eventData = {
        title: $('#title').val(),
        description: $('#description').val(),
        start_date: $('#start_date').val(),
        end_date: $('#end_date').val()
    };
    // Otros campos del evento si es necesario
});
```

Figura 20 Añadir evento

Fragmento de código que permite cuando hago clic en un día concreto en el calendario obtener los datos de ese día, función dateClick.

```
1 reference | 0 overrides
public function searchTeachers(Request $request) { //Función para buscar nuevos profesores
    // Inicializar la consulta para obtener usuarios con el rol de 'teacher'
    $teachers = User::where('rol', 'teacher');

    // Filtrar por nombre de usuario si se proporciona en la solicitud
    if ($request->has('name')) {
        $teachers->where('name', 'like', '%'.$request->name.'%');
    }
    // Filtrar por asignatura si se proporciona en la solicitud
    if ($request->has('subject')) {
        $teachers->whereHas('subjects', function($query) use ($request) {
            $query->where('name', 'like', '%'.$request->subject.'%');
        });
    }
    // Filtrar por nivel si se proporciona en la solicitud
    if ($request->has('level')) {
        $teachers->whereHas('subjects', function($query) use ($request) {
            $query->where('level', 'like', '%'.$request->level.'%');
        });
    }
}
```

Figura 21 Filtro profesores

Este fragmento de código es parte de la función searchTeachers() que permite al alumno buscar profesores utilizando diferentes filtros.

```
</tr>
</thead>
<tbody>
    @foreach ($users as $user)
        <tr role="row" class="odd">
            <td><a href="{{ route('conversation', ['userId' => $user->id]) }}">{{ $user->name }}</a></td>
            <td>{{ $user->rol }}</td>
            @if ($user->rol == 'teacher')
                <td>
                    @foreach ($user->subjects as $subject)
                        {{ $subject->name }}
                        {{ $subject->level }}
                    @endforeach
                </td>
            @else
                <td-></td>
            @endif
        </tr>
    @endforeach
```

Figura 22 Todos los usuarios

Este fragmento de código es parte de la vista chat.blade.php, donde se muestra una tabla con todos los usuarios registrados mediante un bucle foreach. Al hacer clic en un usuario, se accede a otra vista que permite iniciar una conversación con él.