Tecnológico Superior de Jalisco

Tecnológico Superior de Jalisco

Informe Del Proyecto Final (EduShare)

Gestión de Proyectos de Software

Nombre: Alvaro Orozco Pérez

Carrera: ISIC 6°

Nombre De la Maestra: Rosa María Chávez Camarena

SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

JALISCO INNOVACIÓN

Contenido

Introducción

Transformando la Educación a través de la Colaboración y el Acceso Equitativo a Recursos Educativos de Calidad

En un mundo cada vez más digitalizado y conectado, el acceso a una educación de calidad es más crucial que nunca para el éxito personal y profesional. Sin embargo, la disparidad en el acceso a recursos educativos y la falta de colaboración entre estudiantes pueden obstaculizar el aprendizaje y limitar las oportunidades de crecimiento académico. Para abordar estos desafíos y promover un entorno educativo más inclusivo y colaborativo, surge EduShare, una plataforma en línea diseñada para revolucionar la forma en que los estudiantes universitarios acceden comparte y colaboran en recursos educativos.

EduShare no es simplemente una plataforma, sino un movimiento que busca democratizar el acceso a la educación al proporcionar a los estudiantes una plataforma centralizada donde puedan compartir y acceder a una amplia gama de recursos educativos de alta calidad. Desde apuntes y presentaciones hasta libros recomendados y enlaces útiles, EduShare actúa como un banco de recursos colaborativo, fomentando la colaboración entre pares y facilitando un intercambio de conocimientos sin fronteras.

Lo que distingue a EduShare es su enfoque en la comunidad y la colaboración estudiantil. Al crear perfiles personalizados, los estudiantes pueden conectarse con otros con intereses similares, compartir sus propios recursos educativos y beneficiarse de los materiales compartidos por otros. Esta colaboración no solo enriquece la experiencia de aprendizaje de los usuarios, sino que también promueve una cultura de apoyo mutuo y crecimiento compartido en la comunidad académica.

Además de fomentar la colaboración, EduShare se compromete a garantizar un acceso equitativo a recursos educativos de calidad. La plataforma está diseñada para eliminar barreras geográficas y socioeconómicas al ofrecer una amplia variedad de materiales de estudio de forma gratuita o a precios asequibles. De esta manera, EduShare se convierte en un igualador de oportunidades, asegurando que todos los estudiantes, independientemente de su ubicación o situación financiera, tengan acceso a recursos educativos valiosos que enriquezcan su aprendizaje y les ayuden a alcanzar su máximo potencial.

En resumen, EduShare representa mucho más que una plataforma de software; es un movimiento que busca transformar la educación al fomentar la colaboración, promover el acceso equitativo a recursos educativos de calidad y crear una comunidad virtual donde los estudiantes se apoyen mutuamente en su viaje académico. Con su enfoque innovador y su compromiso con la excelencia educativa, EduShare está preparado para redefinir la forma en que los estudiantes acceden y comparten conocimientos en el siglo XXI.

Objetivos

Objetivo General

Desarrollo del software para facilitar el intercambio de conocimientos y recursos educativos entre estudiantes universitarios

Objetivos Específicos

1. Incrementar la disponibilidad de recursos educativos: Aumentar la cantidad y diversidad de materiales educativos disponibles en la plataforma mediante la participación de los usuarios en la subida de recursos, asegurando así un amplio abanico de opciones para todos los usuarios.
2. Mejorar la calidad de los recursos: Implementar sistemas de calificación y comentarios que permitan a los usuarios evaluar la calidad de los recursos educativos, promoviendo la identificación de materiales relevantes y confiables.
3. Promover la participación de los usuarios: Fomentar la participación de los usuarios en la plataforma mediante la gamificación, incentivos y eventos interactivos que estimulen la contribución de recursos, interacción en foros y colaboración en proyectos.
4. Personalizar la experiencia del usuario: Desarrollar algoritmos de recomendación avanzados que utilicen el historial de descargas, calificaciones y preferencias de los usuarios para ofrecer recomendaciones personalizadas y relevantes, adaptadas a las necesidades individuales de cada usuario.
5. Fortalecer la comunidad educativa: Facilitar la creación de conexiones entre usuarios con intereses y áreas de estudio similares, promoviendo la formación de grupos de estudio, debates y colaboraciones académicas que enriquezcan la experiencia de aprendizaje de los usuarios.
6. Garantizar la integridad y autenticidad del contenido: Implementar un sistema de moderación y reporte que permita a los usuarios denunciar contenido inapropiado o de baja calidad, asegurando así la calidad y autenticidad del material disponible en la plataforma.
7. Facilitar el acceso desde dispositivos móviles: Optimizar la plataforma para su uso en dispositivos móviles, asegurando que los usuarios puedan acceder a los recursos educativos en cualquier momento y lugar, promoviendo así un aprendizaje flexible y adaptable a las necesidades individuales de los usuarios.
8. Medir el impacto y la satisfacción de los usuarios: Realizar análisis periódicos de datos para evaluar el impacto de la plataforma en el aprendizaje y la satisfacción de los usuarios, identificando áreas de mejora y oportunidades de crecimiento continuo.

Justificación

EduShare se fundamenta en la creencia de que el acceso a la educación de calidad y el intercambio de conocimientos son fundamentales para el crecimiento académico y personal de los estudiantes universitarios. La plataforma se ha creado con la misión de democratizar el acceso a recursos educativos de alta calidad, eliminando barreras geográficas y socioeconómicas para garantizar que todos los estudiantes tengan igualdad de oportunidades en su aprendizaje. Al facilitar la colaboración entre estudiantes y proporcionar un espacio donde puedan compartir y beneficiarse mutuamente de una amplia variedad de materiales de estudio, EduShare busca no solo enriquecer la experiencia educativa de los usuarios, sino también fomentar una cultura de apoyo mutuo a distancia con mensajes y crecimiento compartido entre los estudiantes universitarios, ahorita solo de Arandas pero escalabre cuanto más valla funcionando. Actualmente, EduShare se destaca por su enfoque en la colaboración estudiantil y el acceso equitativo a recursos educativos de alta calidad. Sin embargo, algunas áreas de mejora podrían incluir la implementación de características adicionales para promover una participación más activa de la comunidad, como la integración de foros de discusión para facilitar la interacción entre usuarios y la organización de eventos en línea, como webinars y charlas, para enriquecer aún más el intercambio de conocimientos. Además, podría mejorar su funcionalidad de búsqueda avanzada mediante la inclusión de filtros más específicos y algoritmos de recomendación más precisos. Además, una mayor atención a la accesibilidad y usabilidad, especialmente en dispositivos móviles, podría ampliar su alcance y hacer que los recursos educativos sean aún más accesibles para todos los usuarios.

Calendario de actividades y descripción

Actividades		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Requerimientos		29 Ene-02 feb	05 Feb-09 Feb	12 Feb-16 feb	19 Feb-23 Feb	26 Feb-01 Mar	04 Mar-08Mar	11 Mar-15 Mar	18 Mar-22 Mar	25 Mar-29 Mar	01 Abril-05 Ma	08 Abril-12 Abr	15 Abril-19 Abr	22 Abril-26 Abr	29 Abril-03 Ma	06 May-10 May	13 May-17 May	20 May-24 may
Entrevista	P																	
	R																	
Contrato	P																	
	R																	
Diagramas	P																	
	R																	
Cotizacion (alcance, tiempo, costos)	P																	
	R																	
Planeación y desarrollo																		
Diseño de la base de datos	P																	
	R																	
Desarrollo de Front-End	P																	
	R																	
Desarrollo de Back-End	P																	
	R																	
Pueba																		
Caja blanca	P																	
	R																	
Caja Negra	P																	
	R																	
Sobre carga	P																	
	R																	
Implementación																		
Entrega del sistema	P																	
	R																	
Documentación	P																	
	R																	

Imagen 1

1. 29 Ene - 02 Feb: Para entrevistar al cliente y comprender sus necesidades en un proyecto de software, escucha atentamente, haz preguntas claras y toma notas
2. 05 Feb - 09 Feb: Para redactar un contrato para EduShare, claridad y transparencia son esenciales. Especifica claramente los términos de servicio, la propiedad intelectual del contenido subido, la responsabilidad de los usuarios y la resolución de disputas
3. 12 Feb - 16 Feb: Los diagramas esenciales incluirían: un diagrama de flujo del usuario, arquitectura del sistema, base de datos, interfaz de usuario, flujo de proceso y seguridad. Estos diagramas proporcionan una visión clara de la funcionalidad, la estructura y la seguridad del proyecto.
4. 19 Feb - 23 Feb: Para definir los requisitos del proyecto EduShare, realiza sesiones de lluvia de ideas con el equipo para identificar funcionalidades clave y prioridades.
5. 26 Feb - 01 Mar: Desarrolla prototipos de la interfaz de usuario de EduShare para visualizar la navegación y la experiencia del usuario antes de la implementación.
6. 04 Mar - 08 Mar: Crea un plan de desarrollo para EduShare, dividiendo el proyecto en tareas específicas y asignando responsabilidades a los miembros del equipo.
7. 11 Mar - 15 Mar: Comienza la implementación del sistema de EduShare, centrándote en la creación de la estructura de la base de datos y la configuración del servidor.
8. 18 Mar - 22 Mar: Avanza con el desarrollo del front-end de EduShare, trabajando en la interfaz de usuario y la experiencia del usuario en la plataforma.

9. 25 Mar - 29 Mar: Continúa el desarrollo del back-end de EduShare, integrando funcionalidades como la subida de recursos, la búsqueda y la gestión de usuarios.
10. 01 Abril - 05 Abril: Realiza pruebas exhaustivas del sistema de EduShare para identificar y corregir posibles errores o problemas de funcionamiento.
11. 08 Abril - 12 Abril: Implementa medidas de seguridad adicionales en EduShare para proteger los datos de los usuarios y garantizar la integridad del sistema.
12. 15 Abril - 19 Abril: Finaliza el desarrollo de EduShare y realiza pruebas finales para asegurarte de que todas las funcionalidades funcionen correctamente.
13. 22 Abril - 26 Abril: Prepara la plataforma de EduShare para su lanzamiento, creando materiales de marketing y comunicándote con posibles usuarios.
14. 29 Abril - 03 May: Lanza oficialmente EduShare al público, asegurándote de proporcionar soporte técnico y recopilar comentarios para futuras mejoras.
15. 06 May - 10 May: Monitorea el rendimiento de EduShare y recopila datos sobre el uso de la plataforma para evaluar su éxito inicial.
16. 13 May - 17 May: Realiza ajustes y mejoras en EduShare según los comentarios de los usuarios y el análisis de datos.
17. 20 May - 24 May: Planifica futuras actualizaciones y expansiones de EduShare para seguir satisfaciendo las necesidades de la comunidad educativa.

Personas Necesarias

1. **Desarrollador de software (3 personas):**
   - Desarrollar y mantener la plataforma EduShare, implementando nuevas funcionalidades y corrigiendo errores.
   - Colaborar con el equipo de diseño para implementar la interfaz de usuario y mejorar la experiencia del usuario.
   - Asegurar la seguridad y estabilidad del sistema, realizando pruebas exhaustivas y actualizaciones regulares.
2. **Diseñador UX/UI (2 personas):**
   - Diseñar la interfaz de usuario de EduShare para garantizar una experiencia de usuario intuitiva y atractiva.
   - Crear wireframes, prototipos y diseños de pantalla para las diferentes características de la plataforma.
   - Colaborar estrechamente con el equipo de desarrollo para garantizar la implementación exitosa de los diseños.
3. **Gerente de Proyecto (1 persona):**

- Supervisar y coordinar todas las actividades relacionadas con el proyecto EduShare, incluida la asignación de tareas y el seguimiento del progreso.
- Gestionar el presupuesto, los plazos y los recursos del proyecto para garantizar su finalización exitosa.
- Servir como punto de contacto principal para los interesados y partes interesadas del proyecto.

4. **Especialista en Marketing (1 persona)**
   - Desarrollar e implementar estrategias de marketing para promover la plataforma EduShare entre la comunidad estudiantil y académica.
   - Crear contenido promocional, gestionar campañas publicitarias y mantener la presencia en redes sociales de EduShare.
   - Analizar métricas de marketing y retroalimentación de usuarios para ajustar las estrategias según sea necesario.
5. **Especialista en Soporte Técnico (2 personas):**
   - Proporcionar soporte técnico y asistencia a los usuarios de EduShare, respondiendo preguntas, solucionando problemas y gestionando solicitudes de ayuda.
   - Realizar pruebas de usuario y recopilar comentarios para identificar áreas de mejora en la plataforma.
   - Colaborar con el equipo de desarrollo para resolver problemas técnicos y mejorar la experiencia del usuario.

Políticas de Comunicación:

- Reuniones semanales de seguimiento del proyecto para todo el equipo.
- Comunicación diaria a través de una plataforma de mensajería instantánea para coordinar tareas y resolver problemas.
- Informes mensuales de progreso enviados a los interesados y partes interesadas.

Interesados:

- Estudiantes universitarios (usuarios finales)
- Profesores y educadores
- Administradores universitarios
- Inversionistas y patrocinadores potenciales

Lugar donde se realizo

EduShare es un proyecto que puede ser implementado en diversas instituciones educativas, como universidades, escuelas preparatorias o incluso institutos de educación superior. El lugar donde se realizo es en la universidad y en la casa del programador

Descripción del lugar:

Biblioteca: La biblioteca podría ser un lugar central para la plataforma, donde los estudiantes puedan acceder a las computadoras para subir, descargar y buscar recursos educativos. Además, la biblioteca podría organizar talleres y eventos para promover el uso de la plataforma.

- **Aulas de clase:** Los profesores podrían utilizar la plataforma para compartir recursos con sus estudiantes, como presentaciones, apuntes y lecturas recomendadas. Además, los estudiantes podrían usar la plataforma para trabajar en grupo en proyectos y tareas.
- **Laboratorios de computación:** Los laboratorios de computación podrían ser un lugar donde los estudiantes puedan acceder a la plataforma y trabajar en sus proyectos de forma individual o en grupo.
- **Espacios comunes:** La plataforma EduShare también podría estar disponible en espacios comunes de la universidad, como cafeterías, salas de estudio y áreas de descanso. Esto permitiría a los estudiantes acceder a los recursos educativos en cualquier momento y lugar.

Organigrama: Dirección de Tecnologías de la Información (DTI): La DTI sería responsable de la instalación, configuración y mantenimiento de la plataforma.

- **Biblioteca:** La biblioteca sería responsable de la selección y curación de los recursos educativos que se subirán a la plataforma.
- **Profesorado:** Los profesores serían responsables de promover el uso de la plataforma entre sus estudiantes y de proporcionar retroalimentación sobre su eficacia.
- **Estudiantes:** Los estudiantes serían los usuarios principales de la plataforma y serían responsables de subir, descargar y calificar los recursos educativos.

Departamento al que beneficiaría el sistema:

Departamento de Estudios Académicos: El Departamento de Estudios Académicos sería responsable de establecer las políticas y procedimientos para el uso de la plataforma.

- **Departamento de Tutoría y Orientación:** El Departamento de Tutoría y Orientación podría utilizar la plataforma para proporcionar recursos y apoyo a los estudiantes que tienen dificultades con sus estudios.
- **Departamento de Servicios Estudiantiles:** El Departamento de Servicios Estudiantiles podría utilizar la plataforma para promover el desarrollo de habilidades de autoaprendizaje y gestión del tiempo entre los estudiantes.

Alcances y Limitaciones

Alcances:

- **Módulo de login:**
  - Permite a los usuarios ingresar al sistema utilizando su nombre de usuario y contraseña.
  - Valida la autenticidad de las credenciales del usuario.
  - Presenta una interfaz amigable y atractiva para el login.
- **Interfaz para el usuario:**
  - Ofrece una interfaz visual agradable y atractiva para los usuarios.
  - Es intuitiva y fácil de usar, permitiendo a los usuarios familiarizarse rápidamente con la plataforma.
  - Utiliza colores que contrastan adecuadamente para mejorar la legibilidad y accesibilidad.
- **Registro para nuevos usuarios:**
  - Permite a cualquier usuario registrarse en la plataforma.
  - El proceso de registro es rápido y sencillo, requiriendo solo unos pocos clics desde la página de inicio.
  - Garantiza la seguridad de los datos ingresados por los usuarios durante el proceso de registro.
- **Módulo para subir archivos:**
  - Proporciona un botón para seleccionar y subir archivos a la plataforma.
  - Permite subir archivos de diversos tipos, como documentos, presentaciones e imágenes.
  - Presenta una interfaz clara y sencilla para el proceso de subida de archivos.
- **Interfaz para moderador:**
  - Otorga al moderador la capacidad de controlar qué contenido se puede subir a la plataforma.
  - Proporciona al moderador acceso a todo el contenido subido por los usuarios.
  - Facilita al moderador eliminar, modificar o mostrar cualquier contenido de la plataforma.
- **Sistema de búsqueda de archivos:**

- Implementa un sistema de búsqueda eficiente que permite a los usuarios encontrar fácilmente los archivos que necesitan.
- Admite búsquedas utilizando expresiones regulares para mayor precisión.
- Permite búsquedas por etiquetas asignadas a los archivos para una mejor organización.

- **Reportes:**
  - Se pueden reportar archivos que no van de acuerdo a la plataforma
  - Se evaluarán de forma objetiva y no estará el control a todos los usuarios

Limitaciones:

- **Módulo de login:**
  - No implementa la recuperación de contraseña en caso de olvido.
  - No incorpora autenticación de dos factores para mayor seguridad.
- **Interfaz para el usuario:**
  - No se ha realizado un estudio de usabilidad para garantizar una experiencia óptima para todos los usuarios.
  - No se ha considerado la accesibilidad para usuarios con discapacidades.
- **Registro para nuevos usuarios:**
  - No se ha implementado la verificación por correo electrónico para confirmar la identidad de los usuarios.
  - No se han establecido políticas de privacidad claras para el manejo de datos de usuarios.
- **Módulo para subir archivos:**
  - No se ha establecido un límite de tamaño para los archivos que se pueden subir.
  - No se ha implementado un sistema de control de versiones para los archivos.
- **Interfaz para moderador:**
  - No se ha definido un proceso claro para la resolución de disputas relacionadas con la moderación de contenido.
  - No se ha implementado un sistema de apelaciones para usuarios que no estén de acuerdo con las decisiones del moderador.
  - No permite al moderador realizar todas las acciones que puede realizar un usuario común.
- **Sistema de búsqueda de archivos:**
  - No se ha implementado la búsqueda por metadatos adicionales, como autor, fecha de creación o palabras clave.

- No se ha considerado la optimización del rendimiento para búsquedas con grandes volúmenes de datos.

- **Etiquetado de archivos:**
  - No se ha establecido un sistema de control de vocabulario para las etiquetas utilizadas.
  - No se ha implementado la sugerencia automática de etiquetas para facilitar el proceso de etiquetado.

Marco Teórico

Programación: El proceso de crear instrucciones detalladas para que una computadora las ejecute y complete una tarea específica. Implica el uso de lenguajes de programación para definir la lógica, el flujo de control y las estructuras de datos del programa.

Ingeniería de software: La aplicación de principios de ingeniería para el diseño, desarrollo, implementación y mantenimiento de software. Se enfoca en crear software de alta calidad, confiable, escalable y seguro que satisfaga las necesidades de los usuarios.

Metodologías de desarrollo: Enfoques estructurados para planificar, diseñar, desarrollar, probar y mantener software. Existen diversas metodologías, como Agile, Scrum, Waterfall y Kanban, cada una con sus propias características y beneficios.

Gestión de proyectos: El proceso de planificar, organizar, dirigir y controlar los recursos para lograr los objetivos de un proyecto de software de manera eficiente y efectiva. Implica la gestión del tiempo, el presupuesto, el alcance, la calidad y los riesgos del proyecto.

Pruebas de software: El proceso de evaluar un software para identificar y corregir errores, verificar que cumple con los requisitos y garantizar que funcione correctamente. Implica diferentes tipos de pruebas, como pruebas unitarias, pruebas de integración, pruebas de sistemas y pruebas de aceptación.

Despliegue de software: El proceso de distribuir e instalar software en un entorno de producción para que los usuarios finales puedan usarlo. Implica la preparación del entorno, la transferencia de archivos, la configuración y las pruebas finales.

Tecnologías: Los componentes y herramientas que se utilizan para desarrollar, implementar y ejecutar software. Incluyen lenguajes de programación, frameworks, bibliotecas, bases de datos, sistemas operativos, servicios en la nube y herramientas de seguridad.

Lenguajes de programación: Herramientas para escribir instrucciones detalladas que la computadora pueda entender. Existen diferentes tipos de lenguajes, como lenguajes de programación de propósito general (Java, Python, C++), lenguajes front-end (JavaScript, HTML, CSS), lenguajes back-end (PHP, Node.js, Python) y lenguajes de scripting (Bash, Python, JavaScript).

Frameworks: Estructuras que proporcionan una base predefinida para desarrollar aplicaciones, simplificando el proceso y promoviendo buenas prácticas de desarrollo. Ejemplos: React, Angular, Vue.js para desarrollo web front-end; Django, Ruby on Rails, Spring Boot para desarrollo web back-end.

Bibliotecas: Colecciones de código precompilado que proporcionan funcionalidades específicas para ahorrar tiempo y esfuerzo en el desarrollo de software. Ejemplos: jQuery, Bootstrap para desarrollo web front-end; NumPy, Pandas para análisis de datos en Python.

Bases de datos: Sistemas para almacenar, organizar y recuperar datos de manera eficiente. Existen diferentes tipos de bases de datos, como bases de datos relacionales (MySQL, PostgreSQL), bases de datos NoSQL (MongoDB, Cassandra) y bases de datos en la nube (Amazon DynamoDB, Microsoft Azure Cosmos DB).

Sistemas operativos: Software que gestiona los recursos de hardware y proporciona una plataforma para ejecutar aplicaciones. Ejemplos: Windows, macOS, Linux, Android, iOS.

Nube: Un modelo de computación donde los recursos informáticos (servidores, almacenamiento, redes) se proporcionan a través de Internet como un servicio. Permite acceder a estos recursos de manera flexible, escalable y bajo demanda. Ejemplos: Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP).

Seguridad informática: La práctica de proteger sistemas, redes y datos de accesos no autorizados, uso, divulgación, disrupción, modificación o destrucción no intencionada. Implica la implementación de medidas de seguridad como firewalls, antivirus, encriptación y control de acceso.

Aplicaciones: Programas informáticos diseñados para realizar tareas específicas o satisfacer necesidades particulares de los usuarios. Existen diversos tipos de aplicaciones, como:

- **Web:** Aplicaciones que se ejecutan en un navegador web y se acceden a través de Internet. Ejemplos: Gmail, Facebook, Netflix.
- **Móviles:** Aplicaciones diseñadas para dispositivos móviles como smartphones y tablets. Ejemplos: WhatsApp, Instagram, Spotify.
- **Escritorio:** Aplicaciones que se instalan y ejecutan en un ordenador personal. Ejemplos: Microsoft Office, Adobe Photoshop, VLC Media Player.
- **Juegos:** Aplicaciones diseñadas para entretenimiento y ocio. Ejemplos: Fortnite, League of Legends, Minecraft.
- **Inteligencia artificial:** Aplicaciones que utilizan algoritmos para simular la inteligencia humana y realizar tareas como el reconocimiento de imágenes, el procesamiento del lenguaje natural y la toma de decisiones. Ejemplos: Siri, Alexa, Google Translate.

- **Machine learning:** Un subconjunto de la inteligencia artificial que se basa en algoritmos que aprenden de los datos para

Procedimientos

Trabajo en clase

Entrevista a usuarios:

- El equipo de desarrollo establecerá una reunión con los usuarios clave del proyecto.
- Durante la reunión, el equipo de desarrollo formulará preguntas abiertas y claras para obtener información sobre las necesidades de los usuarios en cuanto a funcionalidades, usabilidad y aspectos técnicos.
- Se tomarán notas detalladas durante la entrevista para capturar la información relevante.
- Se analizará la información recopilada para identificar patrones y tendencias en las necesidades de los usuarios.

2. Cotización (alcance, tiempo, costos):

- El equipo de desarrollo analizará la información recopilada en la entrevista a usuarios para definir el alcance del proyecto.
- Se elaborará un cronograma detallado que indique las fechas de inicio y finalización de cada etapa del proyecto.
- Se estimarán los costos asociados a cada etapa del proyecto, incluyendo los recursos humanos, materiales y tecnológicos necesarios.
- Se elaborará una cotización formal que incluya el alcance del trabajo, el cronograma y los costos estimados.

3. Planeación y desarrollo:

- El equipo de desarrollo definirá la arquitectura del sistema, considerando aspectos como la escalabilidad, la seguridad y el rendimiento.
- Se diseñará la interfaz de usuario de la plataforma, utilizando prototipos y herramientas de diseño visual para crear una experiencia de usuario intuitiva y atractiva.
- Se desarrollará el código fuente de la plataforma, utilizando lenguajes de programación, frameworks y bibliotecas adecuados para el proyecto.
- Se implementarán las funcionalidades de la plataforma, siguiendo las especificaciones y requerimientos definidos en la etapa de planeación.

- Se realizarán pruebas unitarias y de integración para verificar el correcto funcionamiento de cada módulo de la plataforma.

4. Diseño de la base de datos:

- Se identificarán las entidades y relaciones que se deben almacenar en la base de datos.
- Se diseñará el modelo de datos, utilizando diagramas de entidad-relación (E-R) para representar la estructura de la base de datos.
- Se definirán las tablas, campos y relaciones en la base de datos.
- Se implementará el modelo de datos en un sistema de gestión de bases de datos (SGBD) adecuado.

5. Desarrollo de Front End:

- Se creará el código HTML para definir la estructura de las páginas web de la plataforma.
- Se usará un framework (tailwind) para estilizar la apariencia de las páginas web, utilizando colores, tipografías, diseños y animaciones.
- Se implementará JavaScript para agregar interactividad a las páginas web, permitiendo a los usuarios interactuar con los elementos y realizar acciones como buscar, cargar archivos, comentar y calificar recursos educativos.
- Se optimizará el rendimiento del front-end para garantizar una carga rápida y fluida de las páginas web.

6. Desarrollo de Back End:

- Se diseñará la arquitectura del back-end, considerando aspectos como la escalabilidad, la seguridad y el rendimiento.
- Se desarrollará el código fuente del back-end, utilizando lenguajes de programación adecuados para el proyecto.
- Se implementarán las funcionalidades del back-end, como la autenticación de usuarios, la gestión de recursos educativos, la búsqueda de recursos, el sistema de comentarios y la interacción con la base de datos.
- Se realizarán pruebas unitarias y de integración para verificar el correcto funcionamiento del back-end.

7. Prueba y corrección de errores:

- Se definirán casos de prueba para cada funcionalidad de la plataforma.
- Se realizarán pruebas manuales y automatizadas para identificar errores en la interfaz de usuario

Entrevista para EduShare

Objetivos y Propósito:

¿Cuál es el objetivo principal de implementar EduShare en su entorno educativo?

Dar la comunicación a los alumnos para compartir información que le haya servido para que otras personas puedan utilizarlo a su favor

¿Qué problemas específicos espera resolver con esta plataforma?

Que sea de forma eficaz la búsqueda para ciertos temas que se ven dentro de una carrera

¿Quiénes serán los principales usuarios de EduShare?

1. Estudiantes
2. Profesores
3. Administradores

¿Cuáles son las características más importantes que los usuarios desearían tener?

Que sean estudiantes universitarios

Requisitos Funcionales:

¿Qué funcionalidades específicas espera que tenga EduShare?

Carga de Recursos, Búsqueda y Filtrado, Calificaciones y Comentarios, Gestión de Etiquetas, Sistema de Recomendación, Perfiles de Usuario, Administración del Sistema, Seguridad y Privacidad, Colaboración y Comunicación

¿Existen requisitos específicos de personalización según las necesidades de su institución?

No solo será una plataforma común para cualquier universidad en la que este interezado

¿Necesitará EduShare integrarse con otros sistemas existentes en su institución?

Es la idea en un futuro, pero por el momento no

¿Hay algún formato de archivo específico que deba ser compatible al cargar recursos?

Alguno s de los que deben de ser obligatorios son .docx, pdf, power point, y archivos editables

Seguridad y Privacidad:

¿Cuáles son sus preocupaciones y requisitos de seguridad y privacidad para los datos de los usuarios y los recursos cargados?

Un login y sistema de autentificación para el usuario

¿Necesitará funciones de control de acceso y autenticación robustas?

Solo login y un registro

¿Cómo imagina la interfaz de usuario de EduShare? ¿Tiene alguna preferencia de diseño o estilo?

Minimalista y en un matis de colores

¿Qué dispositivos (computadoras, tabletas, teléfonos móviles) deben ser compatibles?

Por el momento solo computadoras

¿Cómo le gustaría gestionar y organizar el contenido educativo en la plataforma?

Con un tipo de usuario que será el Moderador, el cual podrá evaluar la información que se subirá

¿Existe algún requisito específico para la categorización de recursos?

- Si
- No

¿Cómo espera que los usuarios colaboren y se comuniquen dentro de la plataforma?

Mediante comentarios y chats, en lo que podrán comunicarse de forma personal como grupal

¿Necesitará funciones como mensajería, foros o chat?

Si, en general todas esas, un foro, un chat individual y comentarios

Escalabilidad y Mantenimiento:

¿Cuántos usuarios anticipa que utilizarán la plataforma a corto y largo plazo?

Iniciaremos con una prueba en el Tec, estamos hablando de un aproximado de 200 personas

¿Cuál es su enfoque para el mantenimiento y las actualizaciones a medida que la plataforma evoluciona?

Para un público mucho más extenso, entonces estamos hablando de una expansión de usuarios al mismo tiempo que se utilizaran

Diseño de Bases de Datos

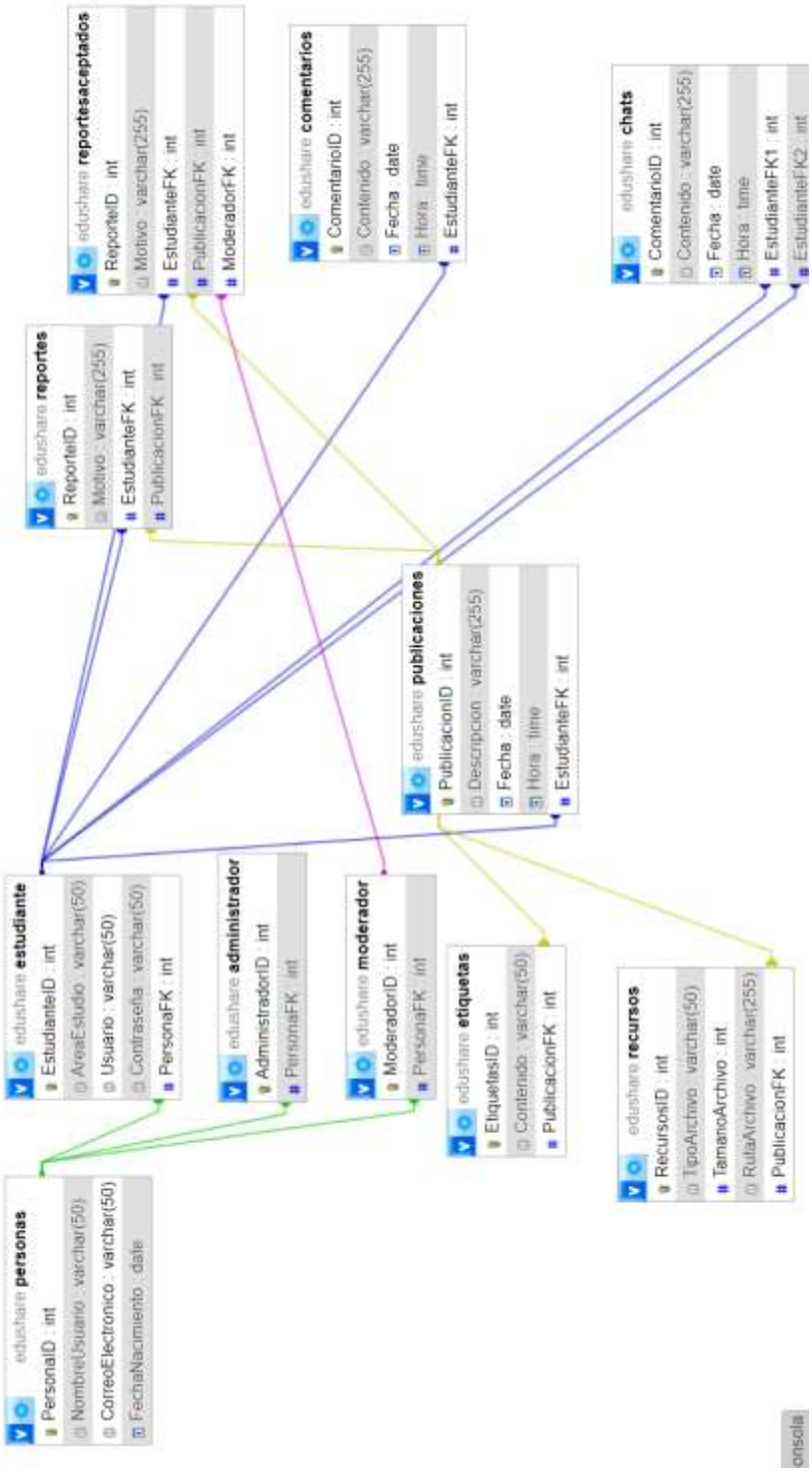
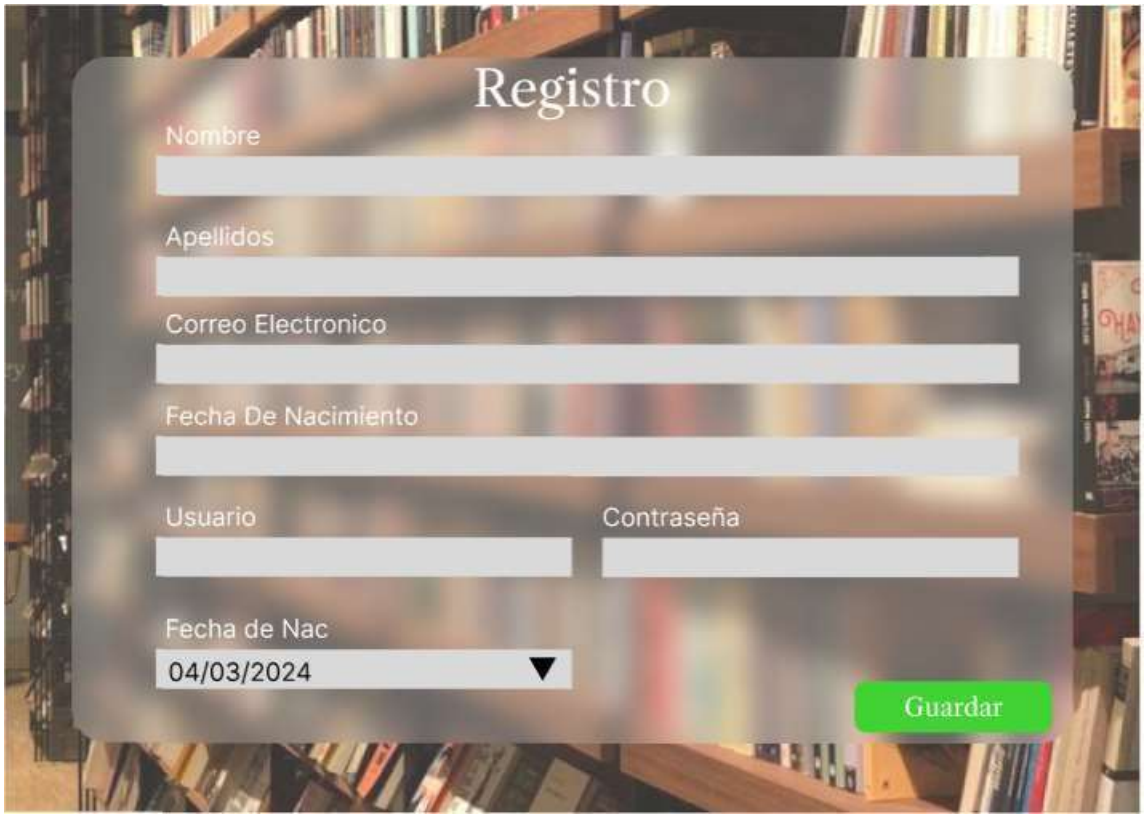


Imagen 2

Diseño de la Interfaz



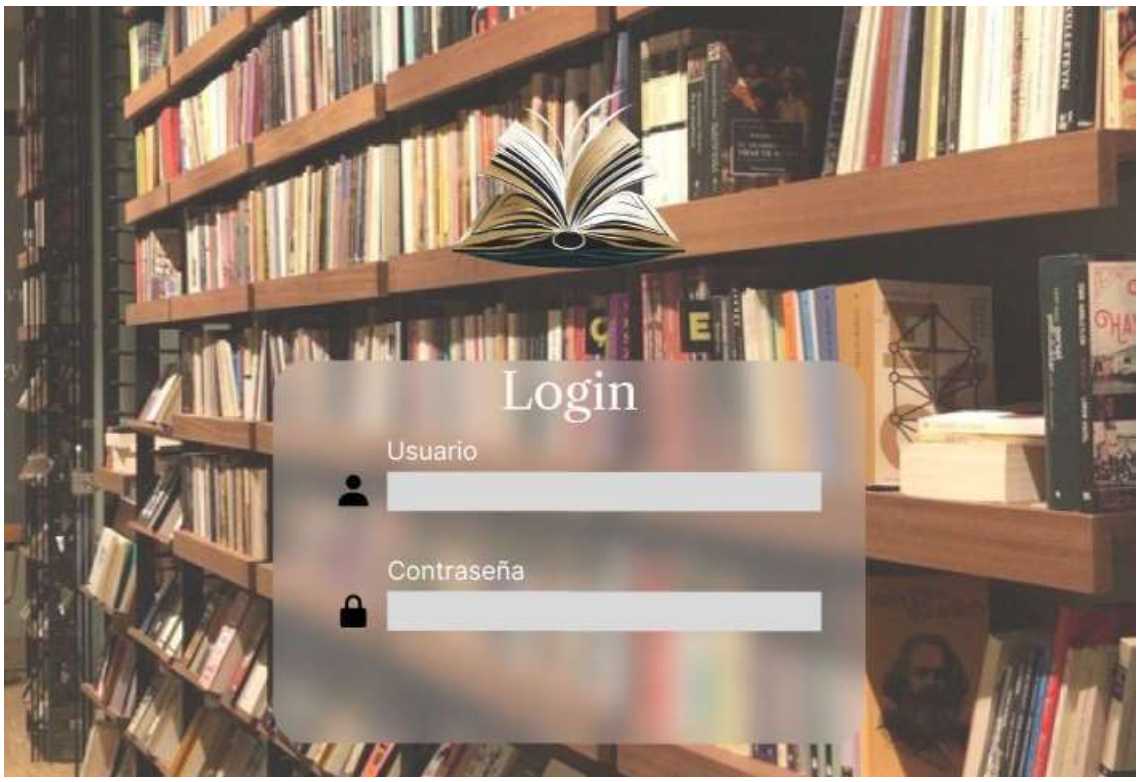


Imagen 3

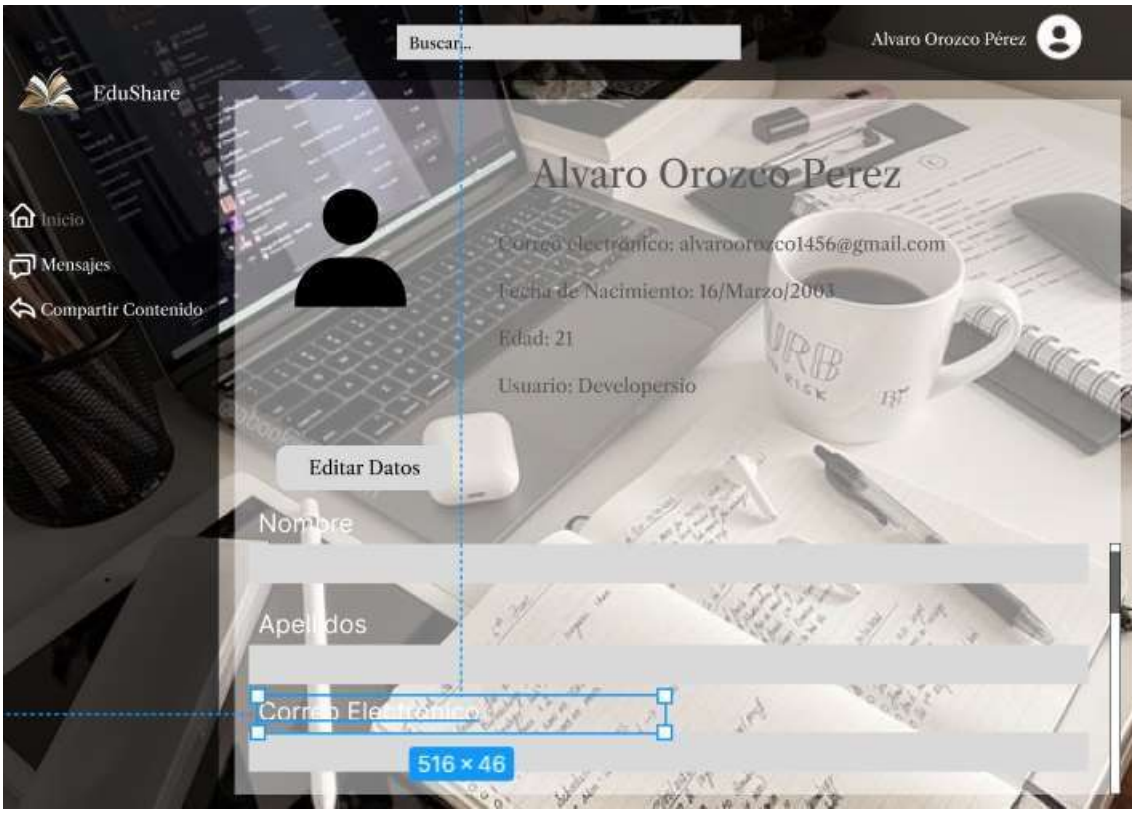


Imagen 5

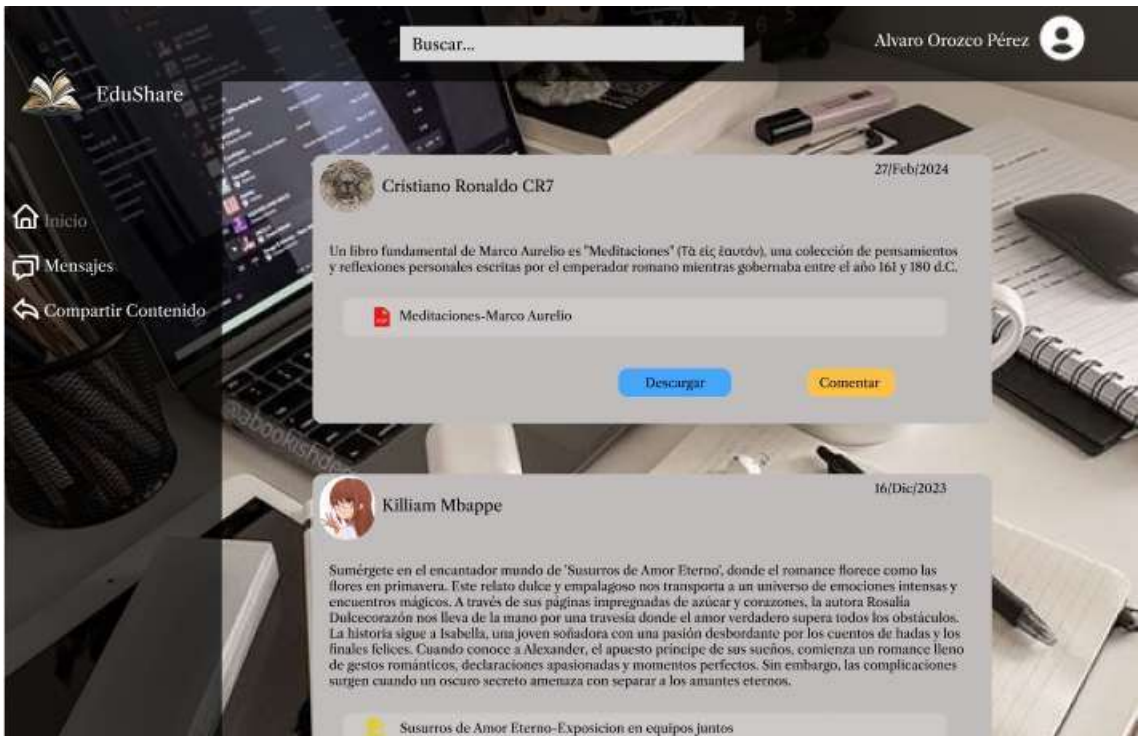


Imagen 4

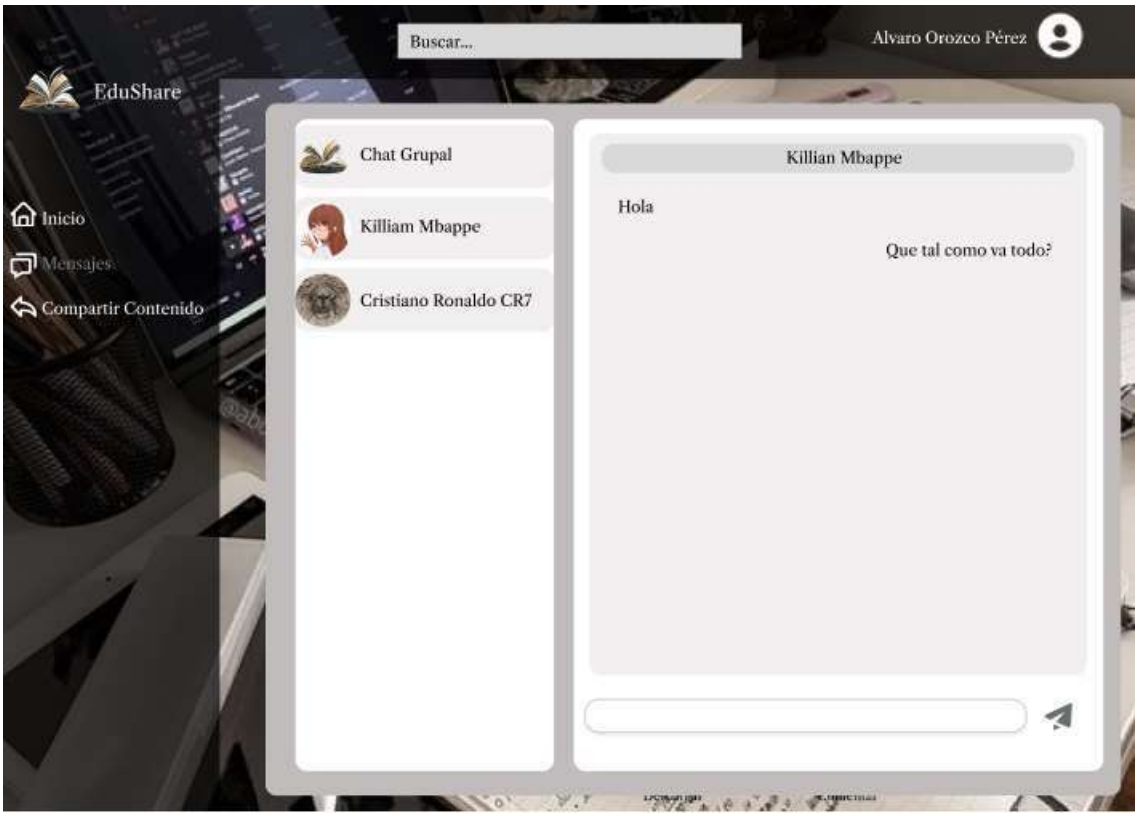


Imagen 7

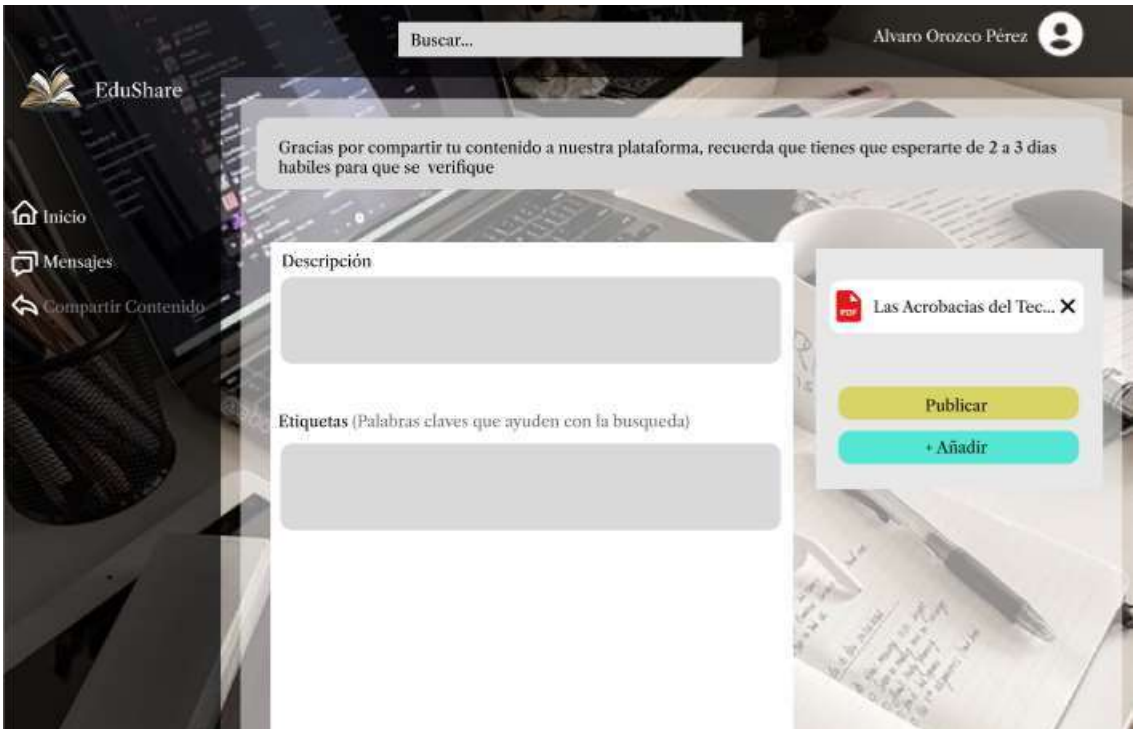


Imagen 6

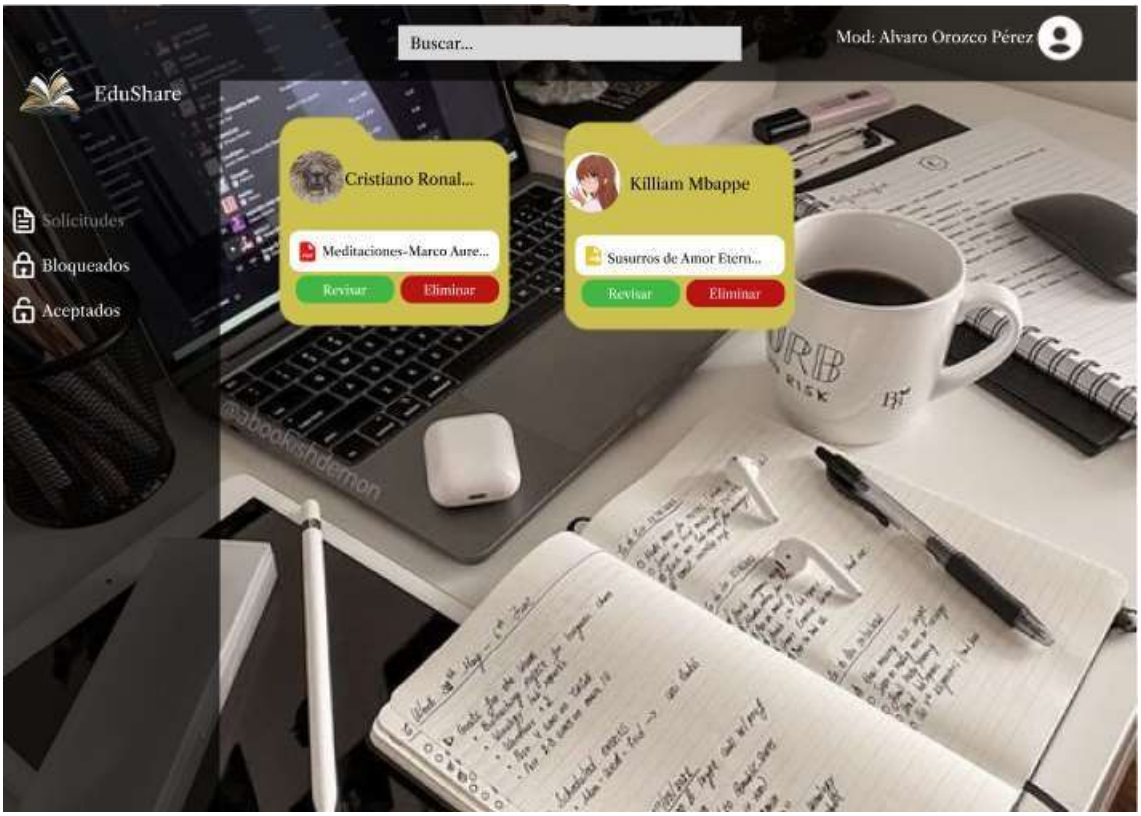


Imagen 10

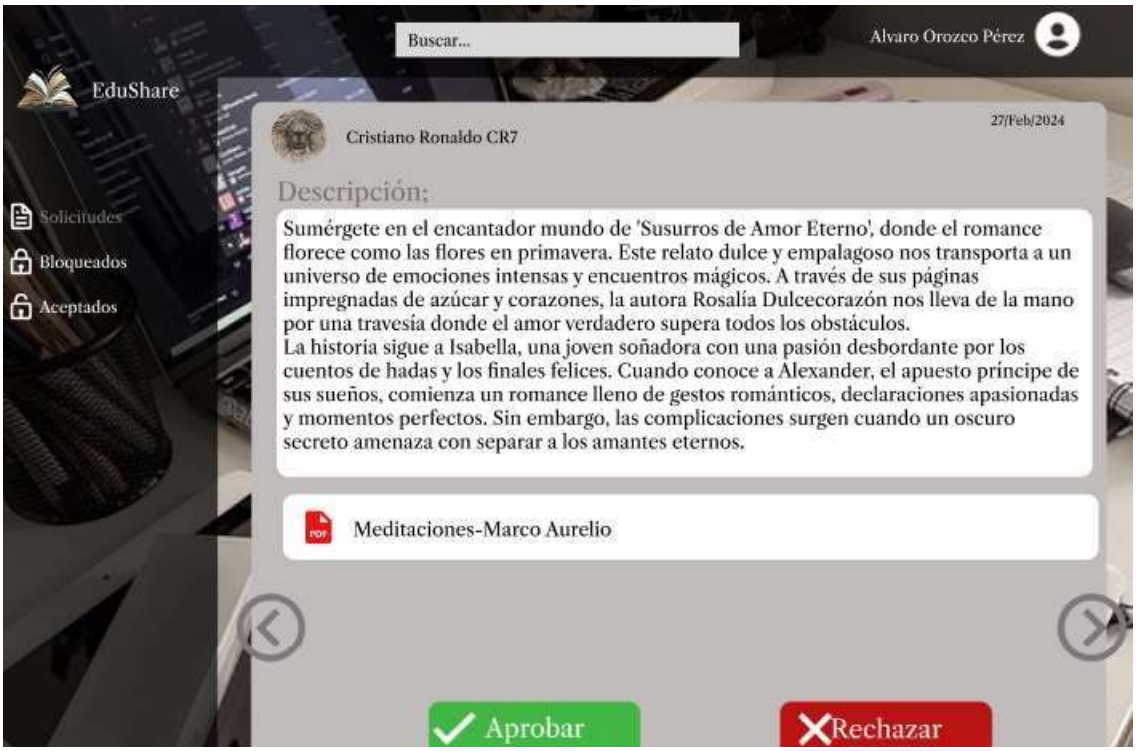


Imagen 9

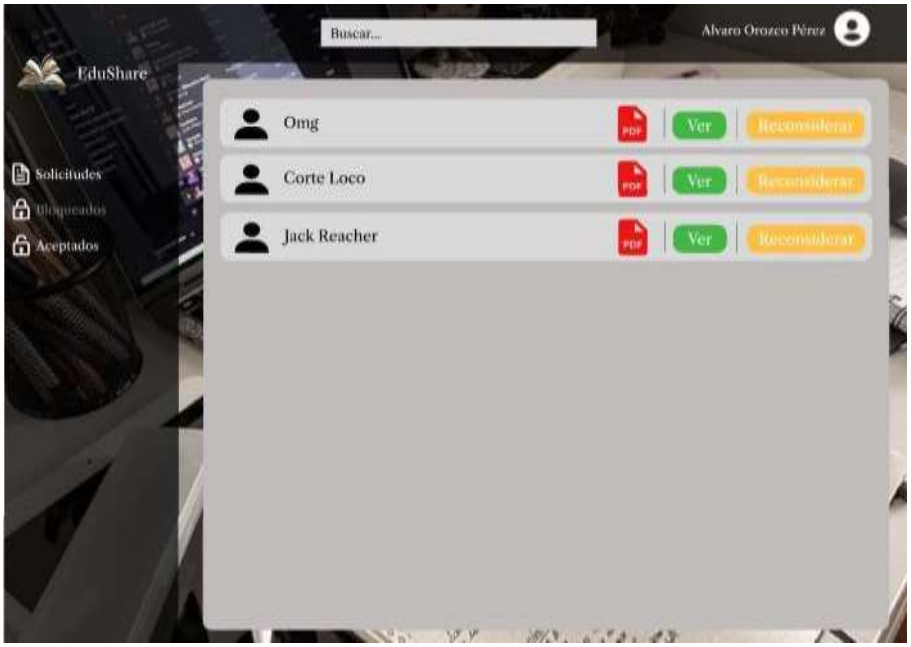


Imagen 11

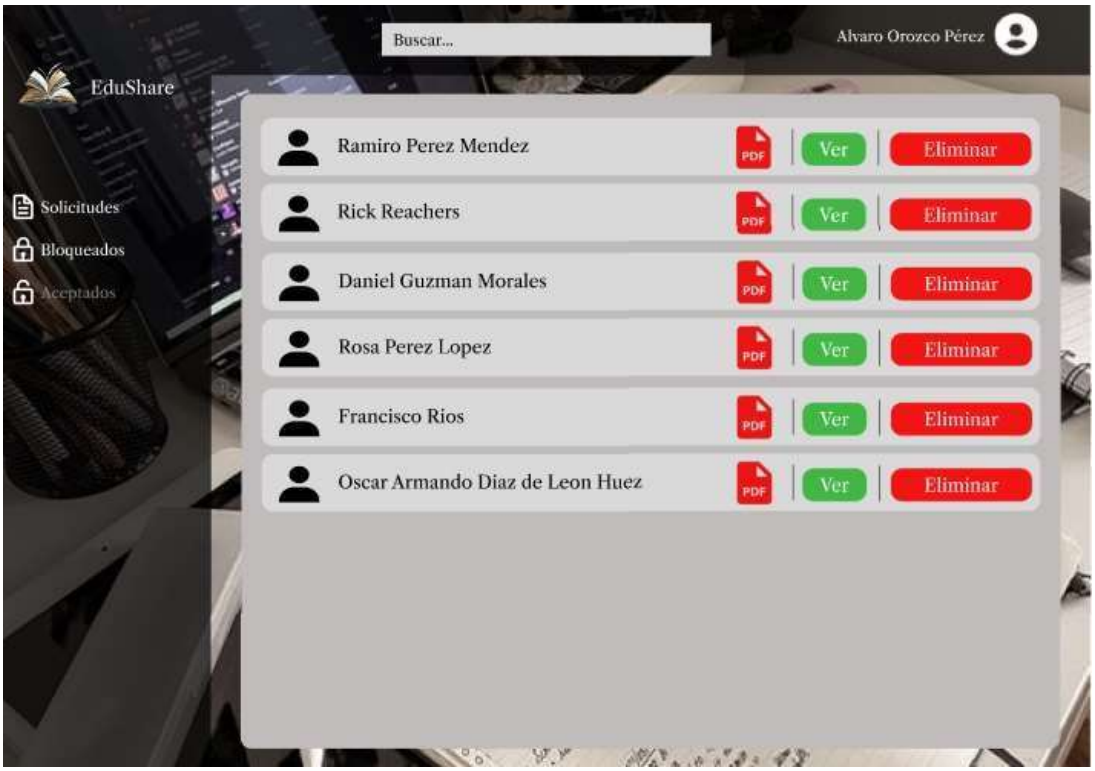


Imagen 12

Creación de la base de datos

```
--

-- Base de datos: `edushare`

--


DELIMITER $$

--

-- Procedimientos

--

CREATE DEFINER=`root`@`localhost` PROCEDURE `InsertarPublicacion` (IN `p_PublicacionID` VARCHAR(100), IN `p_Descripcion` VARCHAR(255), IN `p_Fecha` DATE, IN `p_Hora` TIME, IN `p_EstudianteFK` VARCHAR(100), IN `p_Titulo` VARCHAR(255)) BEGIN


    INSERT INTO publicaciones (PublicacionID, Descripcion, Fecha, Hora, EstudianteFK, Titulo)

    VALUES (p_PublicacionID, p_Descripcion, p_Fecha, p_Hora, p_EstudianteFK, p_Titulo);

END$$


CREATE DEFINER=`root`@`localhost` PROCEDURE `ObtenerPublicacionesOrdenadasPorFecha` () BEGIN

    SELECT * FROM publicaciones ORDER BY Fecha ASC;

END$$


CREATE DEFINER=`root`@`localhost` PROCEDURE `RegistrarPersonaYEstudiante` (IN `p_Nombre` VARCHAR(50), IN `p_Apellido` VARCHAR(50), IN `p_CorreoElectronico` VARCHAR(50), IN `p_FechaNacimiento` DATE, IN `p_AreaEstudio` VARCHAR(50), IN `p_Usuario` VARCHAR(50), IN `p_Contraseña` VARCHAR(50)) BEGIN

    DECLARE v_PersonaID VARCHAR(100);
```

```
    DECLARE v_EstudianteID VARCHAR(100);
    DECLARE v_Estatus INT;
    DECLARE v_Foto VARCHAR(255);

    SET v_PersonaID = UUID();
    SET v_EstudianteID = UUID();

    SET v_Estatus = 1;
    SET v_Foto = NULL;

    INSERT INTO personas (PersonaID, Nombre, Apellidos, CorreoElectronico, FechaNacimiento, Estatus)
    VALUES (v_PersonaID, p_Nombre, p_Apellido, p_CorreoElectronico, p_FechaNacimiento, v_Estatus);

    INSERT INTO estudiante (EstudianteID, AreaEstudio, Usuario, Contraseña, PersonaFK, Estatus, Perfil)
    VALUES (v_EstudianteID, p_AreaEstudio, p_Usuario, p_Contraseña, v_PersonaID, v_Estatus, v_Foto);
END$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `SubirArchivo` (IN `p_descripcion` VARCHAR(255), IN `p_titulo` VARCHAR(255), IN `p_contenido` VARCHAR(255), IN `p_EstudianteActual` VARCHAR(100)) BEGIN
    DECLARE v_fecha DATE;
```

```
    DECLARE v_hora TIME;
    DECLARE v_Id VARCHAR(100);
    DECLARE v_etiqueta VARCHAR(100);

    SET v_fecha = CURDATE();
    SET v_hora = CURTIME();

    SET v_Id = UUID();
    SET v_etiqueta = UUID();

    INSERT INTO publicaciones (PublicacionID, Descripcion, Fecha, Hora, EstudianteFK, Titulo)
    VALUES (v_Id, p_descripcion, v_fecha, v_hora, p_EstudianteActual, p_titulo);

    INSERT INTO etiquetas (EtiquetasID, Contenido, PublicacionFK)
    VALUES (v_etiqueta, p_contenido, v_Id);
END$$

DELIMITER ;

-- ----------------------------------------------------

--
```

```
-- Estructura de tabla para la tabla `actividad`
--

CREATE TABLE `actividad` (
  `ActividadID` int NOT NULL,
  `EstudianteFK1` varchar(100) NOT NULL,
  `EstudianteFK2` varchar(100) NOT NULL,
  `EstadoActividad` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;


-- --------------------------------------------------------

--
-- Estructura de tabla para la tabla `chats`
--

CREATE TABLE `chats` (
  `ComentarioID` int NOT NULL,
  `Contenido` text,
  `FechaHora` datetime DEFAULT NULL,
  `EstudianteFK1` varchar(100) DEFAULT NULL,
  `EstudianteFK2` varchar(100) DEFAULT NULL,
  `ConversacionID` int DEFAULT NULL,
  `Visto` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
-- --------------------------------------------------------

--
-- Estructura de tabla para la tabla `comentarios`
--

CREATE TABLE `comentarios` (
  `ComentarioID` int NOT NULL,
  `Contenido` text NOT NULL,
  `ComentarioFecha` date DEFAULT NULL,
  `ComentarioHora` time DEFAULT NULL,
  `EstudianteFK` varchar(100) DEFAULT NULL,
  `PublicacionFK` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Disparadores `comentarios`
--
DELIMITER $$
CREATE TRIGGER `validar_comentario` BEFORE INSERT ON `comentarios` FOR EACH ROW BEGIN
    DECLARE estudiante_existente INT;

    SELECT COUNT(*) INTO estudiante_existente
    FROM estudiante
    WHERE EstudianteID = NEW.EstudianteFK;
```

```
    IF estudiante_existente = 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'El estudiante asociado al comentario no existe';
    END IF;
END
$$
DELIMITER ;

-- --------------------------------------------------------

--
-- Estructura de tabla para la tabla `estudiante`
--

CREATE TABLE `estudiante` (
  `EstudianteID` varchar(100) NOT NULL,
  `AreaEstudio` varchar(50) DEFAULT NULL,
  `Usuario` varchar(50) DEFAULT NULL,
  `Contraseña` varchar(50) DEFAULT NULL,
  `PersonaFK` varchar(100) DEFAULT NULL,
  `Estatus` int DEFAULT NULL,
  `Perfil` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- --------------------------------------------------------
```

```
--
-- Estructura de tabla para la tabla `etiquetas`
--

CREATE TABLE `etiquetas` (
  `EtiquetasID` int NOT NULL,
  `Contenido` varchar(50) DEFAULT NULL,
  `PublicacionFK` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- --------------------------------------------------------

--
-- Estructura de tabla para la tabla `historial`
--

CREATE TABLE `historial` (
  `HistorialID` int NOT NULL,
  `Motivo` text,
  `Estado` varchar(100) DEFAULT NULL,
  `PublicacionFK` int DEFAULT NULL,
  `ModeradorFK` int DEFAULT NULL,
  `Histofecha` date NOT NULL,
  `Histohora` time NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
-- --------------------------------------------------------

--
-- Estructura de tabla para la tabla `moderador`
--

CREATE TABLE `moderador` (
  `ModeradorID` int NOT NULL,
  `PersonaFK` varchar(100) DEFAULT NULL,
  `contraseña` varchar(255) DEFAULT NULL,
  `status` int DEFAULT NULL,
  `ModPerfil` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;


-- --------------------------------------------------------

--
-- Estructura de tabla para la tabla `personas`
--

CREATE TABLE `personas` (
  `PersonaID` varchar(100) NOT NULL,
  `Nombre` varchar(50) DEFAULT NULL,
  `CorreoElectronico` varchar(50) DEFAULT NULL,
  `FechaNacimiento` date DEFAULT NULL,
```

```
  `Apellidos` varchar(50) NOT NULL,
  `Estatus` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;


-- --------------------------------------------------------

--
-- Estructura de tabla para la tabla `publicaciones`
--

CREATE TABLE `publicaciones` (
  `PublicacionID` int NOT NULL,
  `Descripcion` text,
  `Fecha` date DEFAULT NULL,
  `Hora` time DEFAULT NULL,
  `EstudianteFK` varchar(100) DEFAULT NULL,
  `Titulo` varchar(255) DEFAULT NULL,
  `status` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;


-- --------------------------------------------------------

--
-- Estructura de tabla para la tabla `recursos`
```

```
--

CREATE TABLE `recursos` (
  `RecursosID` int NOT NULL,
  `TipoArchivo` varchar(255) DEFAULT NULL,
  `RutaArchivo` varchar(255) DEFAULT NULL,
  `PublicacionFK` int DEFAULT NULL,
  `ReNombre` varchar(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;


-- --------------------------------------------------------

--
-- Estructura de tabla para la tabla `reportes`
--

CREATE TABLE `reportes` (
  `ReporteID` int NOT NULL,
  `Motivo` text NOT NULL,
  `EstudianteFK` varchar(100) DEFAULT NULL,
  `PublicacionFK` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;


-- --------------------------------------------------------
```

```
--
-- Estructura de tabla para la tabla `reportesaceptados`
--

CREATE TABLE `reportesaceptados` (
  `ReporteID` int NOT NULL,
  `Motivo` varchar(255) DEFAULT NULL,
  `EstudianteFK` varchar(100) DEFAULT NULL,
  `PublicacionFK` int DEFAULT NULL,
  `ModeradorFK` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Disparadores `reportesaceptados`
--
DELIMITER $$
CREATE TRIGGER `registro_actividad_moderador` AFTER INSERT ON `reportesaceptados` FOR EACH ROW BEGIN
    INSERT INTO registro_actividad_moderador (ModeradorID, Accion, Fecha, Hora)
    VALUES (NEW.ModeradorFK, 'Aprobación de reporte', CURDATE(), CURTIME());
END
$$
DELIMITER ;

-- --------------------------------------------------------

--
```

```
-- Estructura Stand-in para la vista `vista_publicaciones`
-- (Véase abajo para la vista actual)
--
CREATE TABLE `vista_publicaciones` (
`PublicacionID` int
,`Descripcion` text
,`Fecha` date
,`Hora` time
,`EstudianteFK` varchar(100)
,`Titulo` varchar(255)
,`status` int
,`RecursosID` int
,`TipoArchivo` varchar(255)
,`RutaArchivo` varchar(255)
,`PublicacionFK` int
,`ReNombre` varchar(200)
,`EstudianteID` varchar(100)
,`AreaEstudio` varchar(50)
,`Usuario` varchar(50)
,`Contraseña` varchar(50)
,`PersonaFK` varchar(100)
,`Estatus` int
,`Perfil` varchar(255)
);

-- --------------------------------------------------
```

--

-- Estructura para la vista `vista_publicaciones`

--

DROP TABLE IF EXISTS `vista_publicaciones`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `vista_publicaciones` AS SELECT `publicaciones`.`PublicacionID` AS `PublicacionID`, `publicaciones`.`Descripcion` AS `Descripcion`, `publicaciones`.`Fecha` AS `Fecha`, `publicaciones`.`Hora` AS `Hora`, `publicaciones`.`EstudianteFK` AS `EstudianteFK`, `publicaciones`.`Titulo` AS `Titulo`, `publicaciones`.`status` AS `status`, `recursos`.`RecursosID` AS `RecursosID`, `recursos`.`TipoArchivo` AS `TipoArchivo`, `recursos`.`RutaArchivo` AS `RutaArchivo`, `recursos`.`PublicacionFK` AS `PublicacionFK`, `recursos`.`ReNombre` AS `ReNombre`, `estudiante`.`EstudianteID` AS `EstudianteID`, `estudiante`.`AreaEstudio` AS `AreaEstudio`, `estudiante`.`Usuario` AS `Usuario`, `estudiante`.`Contraseña` AS `Contraseña`, `estudiante`.`PersonaFK` AS `PersonaFK`, `estudiante`.`Estatus` AS `Estatus`, `estudiante`.`Perfil` AS `Perfil` FROM ((`publicaciones` join `recursos` on((`publicaciones`.`PublicacionID` = `recursos`.`PublicacionFK`))) join `estudiante` on((`publicaciones`.`EstudianteFK` = `estudiante`.`EstudianteID`))) WHERE ((`publicaciones`.`status` = 2) OR (`publicaciones`.`status` = 5)) ORDER BY `publicaciones`.`Fecha` ASC, `publicaciones`.`Hora` ASC ;

--

-- Índices para tablas volcadas

--

--

-- Indices de la tabla `actividad`

--

ALTER TABLE `actividad`

ADD PRIMARY KEY (`ActividadID`),

ADD KEY `EstudianteFK2` (`EstudianteFK2`),

```
  ADD KEY `idx_actividad_estudiante` (`EstudianteFK1`,`EstudianteFK2`);

--
-- Indices de la tabla `chats`
--
ALTER TABLE `chats`
  ADD PRIMARY KEY (`ComentarioID`),
  ADD KEY `EstudianteFK2` (`EstudianteFK2`),
  ADD KEY `ConversacionID` (`ConversacionID`),
  ADD KEY `idx_chats_estudiante_conversacion` (`EstudianteFK1`,`EstudianteFK2`,`ConversacionID`),
  ADD KEY `idx_chats_fechahora` (`FechaHora`);

--
-- Indices de la tabla `comentarios`
--
ALTER TABLE `comentarios`
  ADD PRIMARY KEY (`ComentarioID`),
  ADD KEY `EstudianteFK` (`EstudianteFK`),
  ADD KEY `idx_comentarios_publicacion` (`PublicacionFK`);

--
-- Indices de la tabla `estudiante`
--
ALTER TABLE `estudiante`
  ADD PRIMARY KEY (`EstudianteID`),
  ADD KEY `PersonaFK` (`PersonaFK`),
```

```
  ADD KEY `idx_estudiante_usuario` (`Usuario`);

--
-- Indices de la tabla `etiquetas`
--
ALTER TABLE `etiquetas`
  ADD PRIMARY KEY (`EtiquetasID`),
  ADD KEY `idx_etiquetas_publicacion` (`PublicacionFK`);

--
-- Indices de la tabla `historial`
--
ALTER TABLE `historial`
  ADD PRIMARY KEY (`HistorialID`),
  ADD KEY `idx_historial_publicacion` (`PublicacionFK`),
  ADD KEY `idx_historial_moderador` (`ModeradorFK`);

--
-- Indices de la tabla `moderador`
--
ALTER TABLE `moderador`
  ADD PRIMARY KEY (`ModeradorID`),
  ADD KEY `PersonaFK` (`PersonaFK`);

--
-- Indices de la tabla `personas`
--
```

```
ALTER TABLE `personas`
  ADD PRIMARY KEY (`PersonaID`);

--
-- Indices de la tabla `publicaciones`
--
ALTER TABLE `publicaciones`
  ADD PRIMARY KEY (`PublicacionID`),
  ADD KEY `idx_publicaciones_estudiante` (`EstudianteFK`);

--
-- Indices de la tabla `recursos`
--
ALTER TABLE `recursos`
  ADD PRIMARY KEY (`RecursosID`),
  ADD KEY `idx_recursos_publicacion` (`PublicacionFK`);

--
-- Indices de la tabla `reportes`
--
ALTER TABLE `reportes`
  ADD PRIMARY KEY (`ReporteID`),
  ADD KEY `PublicacionFK` (`PublicacionFK`),
  ADD KEY `idx_reportes_estudiante_publicacion` (`EstudianteFK`,`PublicacionFK`);

--
-- Indices de la tabla `reportesaceptados`
```

```
--

ALTER TABLE `reportesaceptados`
  ADD PRIMARY KEY (`ReporteID`),
  ADD KEY `PublicacionFK` (`PublicacionFK`),
  ADD KEY `idx_reportesaceptados_estudiante_publicacion` (`EstudianteFK`,`PublicacionFK`),
  ADD KEY `idx_reportesaceptados_moderador` (`ModeradorFK`);


--
-- AUTO_INCREMENT de las tablas volcadas
--


--
-- AUTO_INCREMENT de la tabla `actividad`
--
ALTER TABLE `actividad`
  MODIFY `ActividadID` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;


--
-- AUTO_INCREMENT de la tabla `comentarios`
--
ALTER TABLE `comentarios`
  MODIFY `ComentarioID` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=21;


--
-- AUTO_INCREMENT de la tabla `historial`
--
```

ALTER TABLE `historial`

MODIFY `HistorialID` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=37;

--

-- AUTO_INCREMENT de la tabla `reportes`

--

ALTER TABLE `reportes`

MODIFY `ReporteID` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=14;

--

-- Restricciones para tablas volcadas

--

--

-- Filtros para la tabla `actividad`

--

ALTER TABLE `actividad`

ADD CONSTRAINT `actividad_ibfk_1` FOREIGN KEY (`EstudianteFK1`) REFERENCES `estudiante` (`EstudianteID`),

ADD CONSTRAINT `actividad_ibfk_2` FOREIGN KEY (`EstudianteFK2`) REFERENCES `estudiante` (`EstudianteID`);

--

-- Filtros para la tabla `chats`

--

ALTER TABLE `chats`

ADD CONSTRAINT `chats_ibfk_1` FOREIGN KEY (`EstudianteFK1`) REFERENCES `estudiante` (`EstudianteID`),

ADD CONSTRAINT `chats_ibfk_2` FOREIGN KEY (`EstudianteFK2`) REFERENCES `estudiante` (`EstudianteID`);

--

-- Filtros para la tabla `comentarios`

--

ALTER TABLE `comentarios`

ADD CONSTRAINT `comentarios_ibfk_1` FOREIGN KEY (`EstudianteFK`) REFERENCES `estudiante` (`EstudianteID`),

ADD CONSTRAINT `fk_comentarios_publicaciones` FOREIGN KEY (`PublicacionFK`) REFERENCES `publicaciones` (`PublicacionID`);

--

-- Filtros para la tabla `estudiante`

--

ALTER TABLE `estudiante`

ADD CONSTRAINT `estudiante_ibfk_1` FOREIGN KEY (`PersonaFK`) REFERENCES `personas` (`PersonaID`);

--

-- Filtros para la tabla `etiquetas`

--

ALTER TABLE `etiquetas`

ADD CONSTRAINT `etiquetas_ibfk_1` FOREIGN KEY (`PublicacionFK`) REFERENCES `publicaciones` (`PublicacionID`);

--

-- Filtros para la tabla `historial`

--

ALTER TABLE `historial`

ADD CONSTRAINT `historial_ibfk_1` FOREIGN KEY (`PublicacionFK`) REFERENCES `publicaciones` (`PublicacionID`),

ADD CONSTRAINT `historial_ibfk_2` FOREIGN KEY (`ModeradorFK`) REFERENCES `moderador` (`ModeradorID`);

--

-- Filtros para la tabla `moderador`

--

ALTER TABLE `moderador`

ADD CONSTRAINT `moderador_ibfk_1` FOREIGN KEY (`PersonaFK`) REFERENCES `personas` (`PersonaID`);

--

-- Filtros para la tabla `publicaciones`

--

ALTER TABLE `publicaciones`

ADD CONSTRAINT `publicaciones_ibfk_1` FOREIGN KEY (`EstudianteFK`) REFERENCES `estudiante` (`EstudianteID`);

--

-- Filtros para la tabla `recursos`

--

ALTER TABLE `recursos`

ADD CONSTRAINT `recursos_ibfk_1` FOREIGN KEY (`PublicacionFK`) REFERENCES `publicaciones` (`PublicacionID`);

--

-- Filtros para la tabla `reportes`

--

ALTER TABLE `reportes`

ADD CONSTRAINT `reportes_ibfk_1` FOREIGN KEY (`EstudianteFK`) REFERENCES `estudiante` (`EstudianteID`),

ADD CONSTRAINT `reportes_ibfk_2` FOREIGN KEY (`PublicacionFK`) REFERENCES `publicaciones` (`PublicacionID`);

--

-- Filtros para la tabla `reportesaceptados`

--

ALTER TABLE `reportesaceptados`

ADD CONSTRAINT `reportesaceptados_ibfk_1` FOREIGN KEY (`EstudianteFK`) REFERENCES `estudiante` (`EstudianteID`),

ADD CONSTRAINT `reportesaceptados_ibfk_2` FOREIGN KEY (`PublicacionFK`) REFERENCES `publicaciones` (`PublicacionID`),

ADD CONSTRAINT `reportesaceptados_ibfk_3` FOREIGN KEY (`ModeradorFK`) REFERENCES `moderador` (`ModeradorID`);

COMMIT;

Pruebas

Al lo largo de estos dos o tres meses que llevo haciendo este proyecto tuve una cantidad excesiva de errores, por diferentes motivos el principal es ser nuevo en el uso de Node.js que es una tecnología que se adecua a lo que yo quería, cumplió mis expectativas sin embargo no puedo negar que me genero muchos dolores de cabeza

1. Conexión a la Base de Datos

Error: No cerrar las conexiones a la base de datos

- **Problema**: Dejar conexiones abiertas puede agotar los recursos del servidor y provocar que la aplicación se vuelva inestable.
- **Solución**: Usar un pool de conexiones y asegurarse de liberar las conexiones cuando no se necesiten.

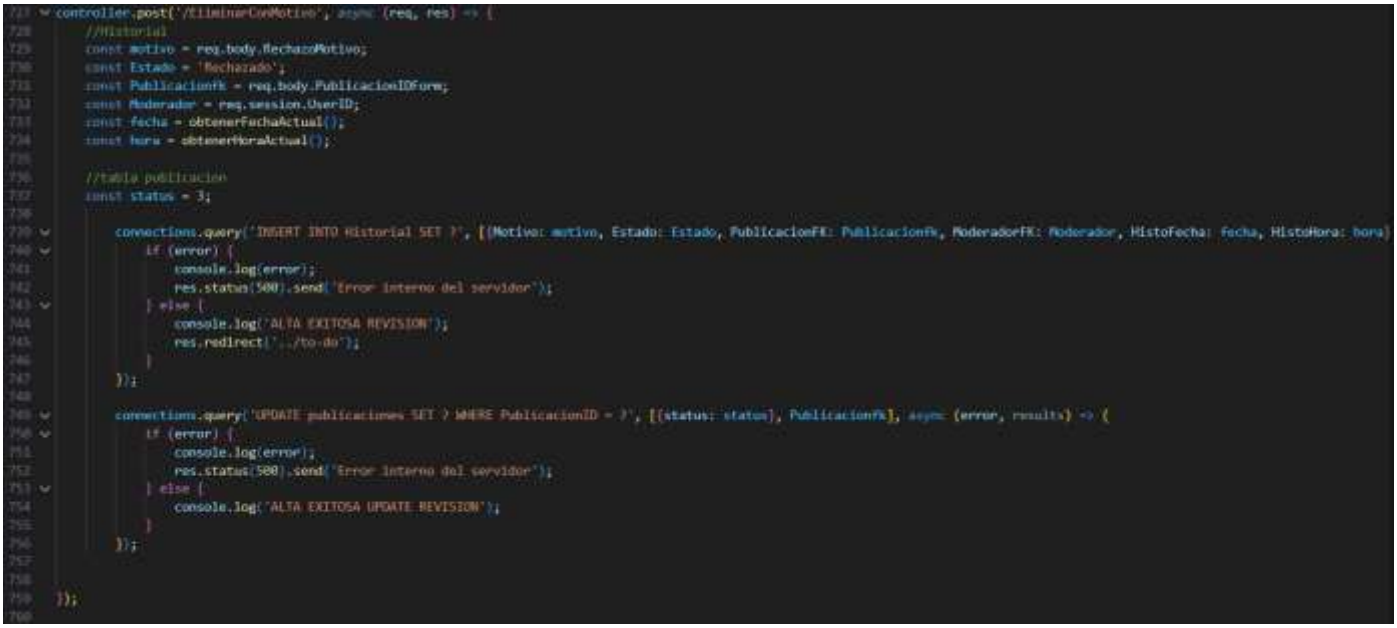
```
const mysql = require ('mysql');
const connection = mysql.createConnection({
    host: 'localhost',
    user: 'root',
    password: '12345678',
    database: 'edushare'
})

connection.connect((error)=>{
    if(error){
        console.log('error de conexion es: '+error);
        return;
    }
    console.log ('¡Conectado a la base de datos!')
})

module.exports = connection;
```

Imagen 13

2. Inyección de SQL

Error: No usar consultas preparadas

- **Problema**: No usar consultas preparadas puede exponer la aplicación a ataques de inyección SQL.
- **Solución**: Siempre usar consultas preparadas para insertar valores en las consultas SQL.



```
controller.post('/EliminarConMotivo', async (req, res) => {
    //Historial
    const motivo = req.body.RechazoMotivo;
    const Estado = 'Rechazado';
    const PublicacionFk = req.body.PublicacionIDForm;
    const Moderador = req.session.UserID;
    const fecha = obtenerFechaActual();
    const hora = obtenerHoraActual();

    //tabla publicacion
    const status = 3;

        connections.query('INSERT INTO Historial SET ?', [{Motivo: motivo, Estado: Estado, PublicacionFK: PublicacionFk, ModeradorFK: Moderador, HistoFecha: fecha, HistoHora: hora}
            if (error) {
                console.log(error);
                res.status(500).send('Error interno del servidor');
            } else {
                console.log('ALTA EXITOSA REVISION');
                res.redirect('../to-do');
            }
        });

        connections.query('UPDATE publicaciones SET ? WHERE PublicacionID = ?', [{status: status}, PublicacionFk], async (error, results) => {
            if (error) {
                console.log(error);
                res.status(500).send('Error interno del servidor');
            } else {
                console.log('ALTA EXITOSA UPDATE REVISION');
            }
        });


});
```

Imagen 14

3. Manejo de Errores

Error: No manejar adecuadamente los errores

- **Problema**: Ignorar o manejar mal los errores puede llevar a caídas de la aplicación y problemas de seguridad.
- **Solución**: Implementar una estrategia robusta de manejo de errores.

```
pool.getConnection((err, connection) => {
    if (err) {
      console.error('Error de conexión a la base de datos:', err);
      return;
    }
    connection.query('SELECT * FROM users', (error, results) => {
      connection.release();
      if (error) {
        console.error('Error en la consulta:', error);
        return;
      }
      // Procesa los resultados
    });
  });

```

Imagen 15

4. Escalabilidad

Error: No considerar la escalabilidad desde el principio

- **Problema**: Diseñar sin pensar en la escalabilidad puede llevar a problemas de rendimiento y mantenimiento a largo plazo.
- **Solución**: Usar técnicas como la paginación, índices apropiados y optimización de consultas.

```
SELECT * FROM users LIMIT 10 OFFSET 0;
```

Imagen 16

5. Optimización de Consultas

Error: Consultas no optimizadas

- **Problema**: Consultas lentas pueden afectar el rendimiento de la aplicación.
- **Solución**: Analizar y optimizar las consultas SQL, usar índices y evitar consultas innecesariamente complejas.

```
4   EXPLAIN SELECT * FROM users WHERE email = 'example@example.com';
5
```

Imagen 17

6. Seguridad

Error: Almacenar contraseñas en texto plano

- **Problema**: Almacenar contraseñas en texto plano puede comprometer la seguridad de los usuarios.
- **Solución**: Usar hashing seguro para contraseñas, como bcrypt.

```
1   const bcrypt = require('bcrypt');
2   const saltRounds = 10;
3   bcrypt.hash(password, saltRounds, (err, hash) => {
4     // Almacenar hash en la base de datos
5   });
6
```

Imagen 18

7. Transacciones

Error: No usar transacciones para operaciones críticas

- **Problema**: No usar transacciones puede llevar a inconsistencias en la base de datos en caso de fallos.
- **Solución**: Usar transacciones para operaciones que requieren atomicidad.

```
pool.getConnection((err, connection) => {
  if (err) throw err;
  connection.beginTransaction(err => {
    if (err) throw err;
    connection.query('INSERT INTO orders SET ?', orderData, (error, results) => {
      if (error) {
        return connection.rollback(() => {
          throw error;
        });
      }
      connection.commit(err => {
        if (err) {
          return connection.rollback(() => {
            throw err;
          });
        }
        console.log('Transaction complete.');
        connection.release();
      });
    });
  });
});
```

Imagen 19

8. Sincronización de Datos

Error: No manejar correctamente la sincronización entre múltiples instancias

- **Problema**: Las aplicaciones distribuidas pueden tener problemas de sincronización de datos.
- **Solución**: Usar mecanismos de sincronización como locks, y considerar bases de datos distribuidas si es necesario.

9. Manejo de Sesiones

Error: No manejar correctamente las sesiones de usuario

- **Problema**: Mal manejo de sesiones puede llevar a problemas de seguridad y experiencia de usuario.
- **Solución**: Usar librerías como **express-session** y almacenarlas de manera segura.

```
const session = require('express-session');
app.use(session({
  secret: 'your-secret-key',
  resave: false,
  saveUninitialized: true,
  cookie: { secure: true }
}));
```

Imagen 20

Resultados

Interfaz y codificación

Login

Es el sistema de autentificación muy sencillo, lo que hacemos es una búsqueda rápida con las credenciales del usuario que quiere iniciar sesión, y los datos importantes los guardamos en variables de sesión

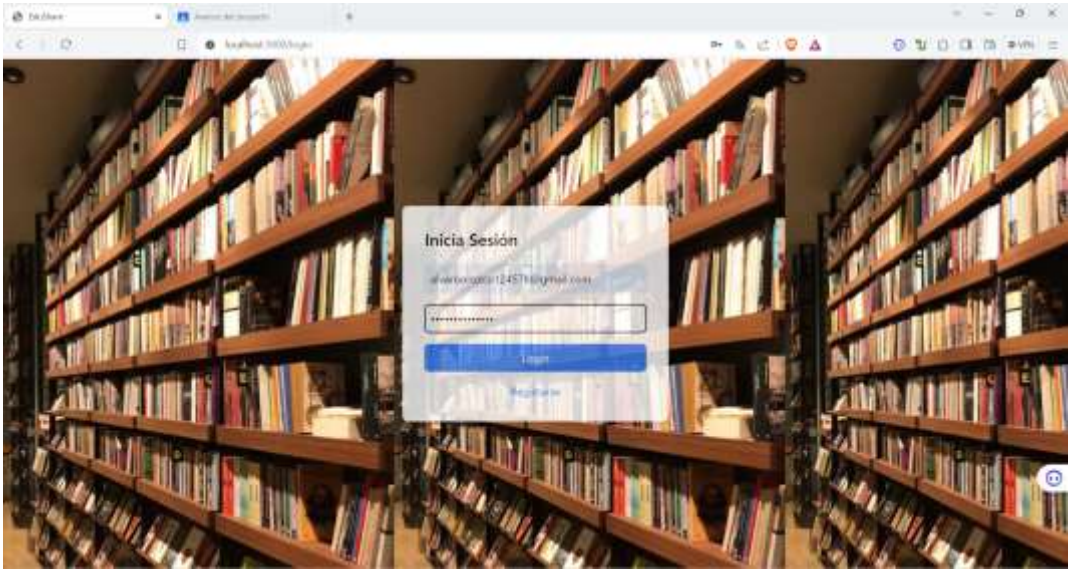


Imagen 21

Registro

Aquí se podrán registrar los nuevos usuarios, como tal es un formulario para agarrar los datos de los usuarios

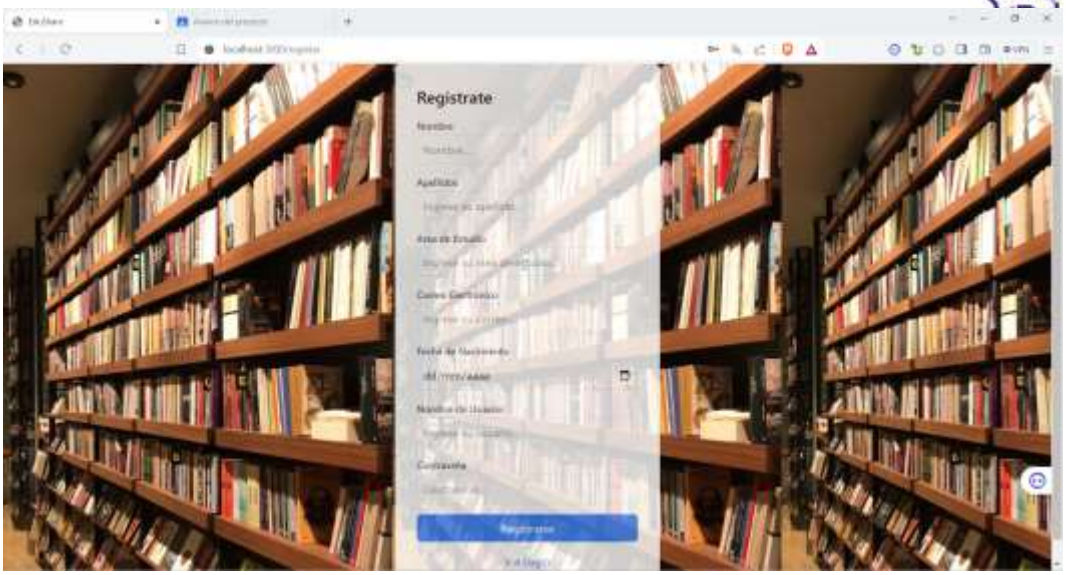


Imagen 22

Inicio

Aquí esta como el Home o el feedback, tratando de imitar un poco como lo que hay en Facebook, esto para ver la información de mejor manera, poderlo descargar, comentar y mandar mensajes (lo de los mensajes y comentarios todavía no lo termino)

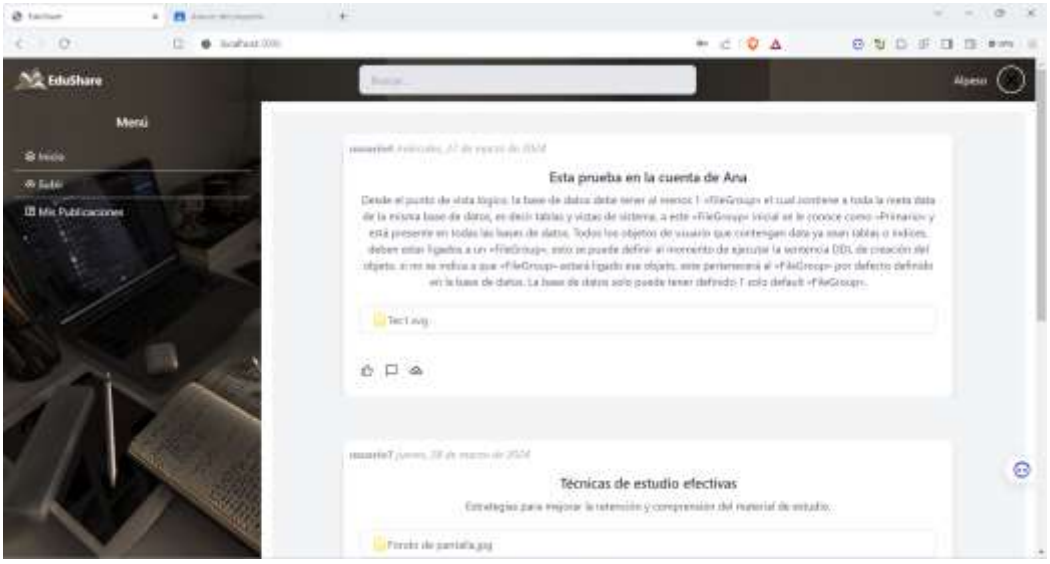


Imagen 23

Subir archivos

Este es el formulario para subir archivos a la plataforma, es un diseño muy básico en el que podemos subir la información necesaria y además es muy intuitivo que se debe de subir

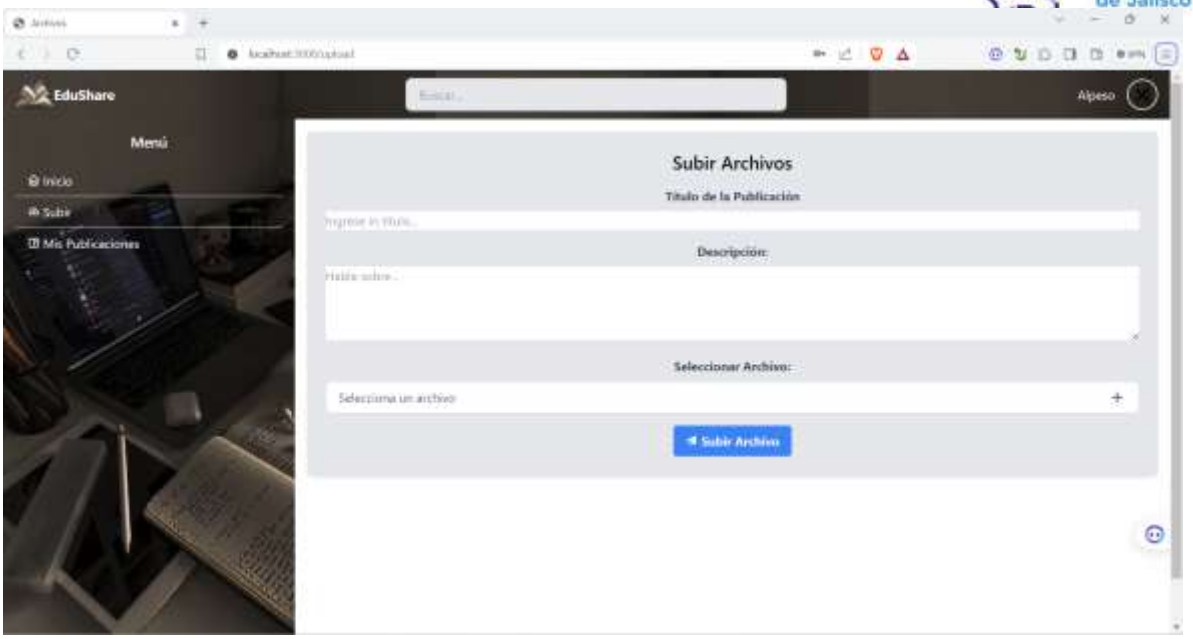

EduShare
Alpeso
Menú
Inicio
Subir
Mis Publicaciones
Subir Archivos
Título de la Publicación
Descripción:
Seleccionar Archivo:
Selecciona un archivo
Subir Archivo

Imagen 24

Mis Publicaciones

Este CRUD está completo y es para la administración de las publicaciones, tiene tres apartados que son En espera: Son aquelas publicaciones recientes que los administradores todavía los tienen en lista de espera, y no ha habido alguna decisión

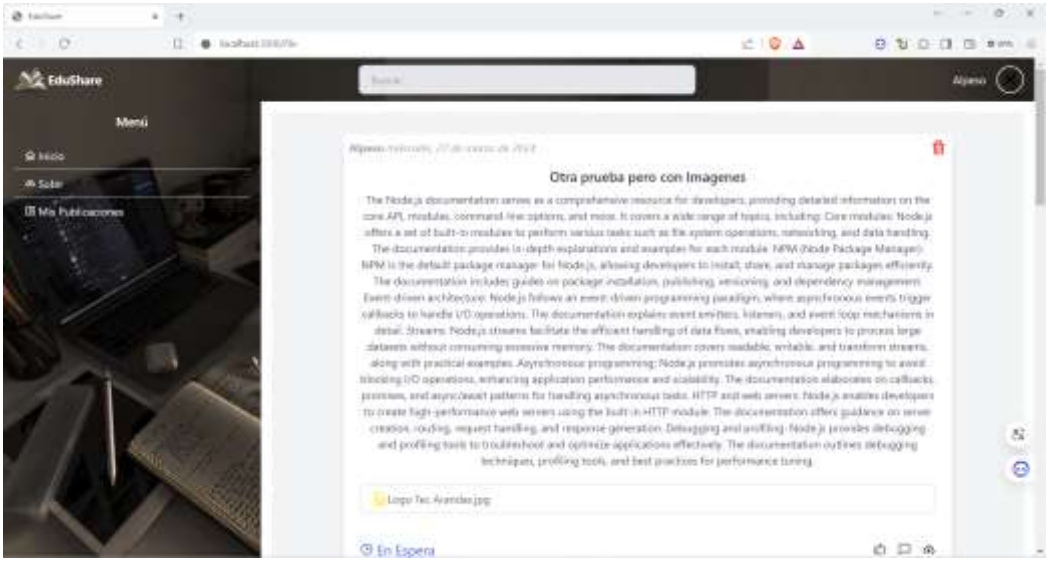


Imagen 25

Aceptados

En esta parte están los que el moderador ya dio un veredicto y están funcionando bien, en esta parte no ha habido problema

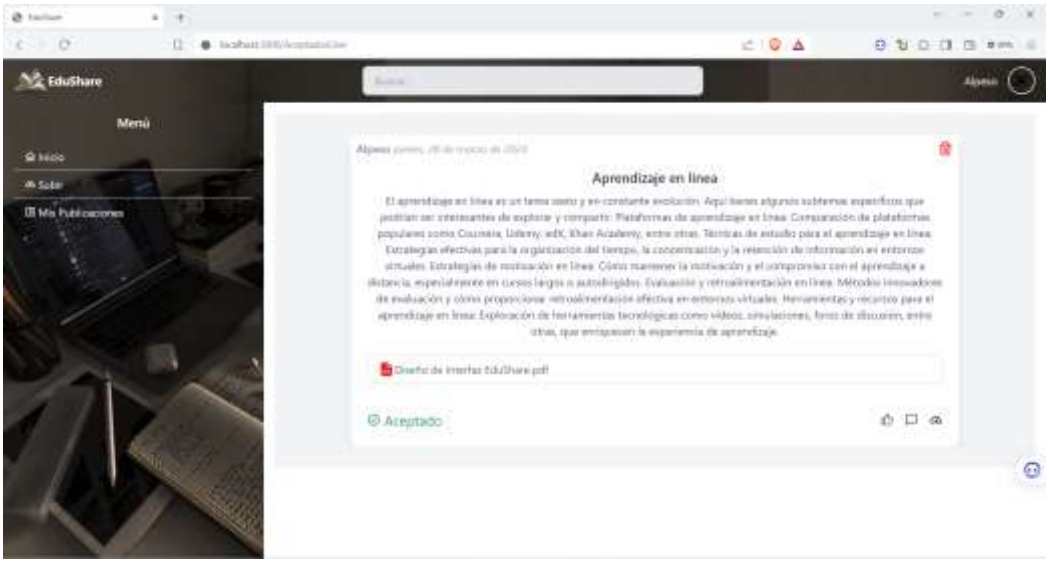


Imagen 26

Rechazados

En esta parte se muestran lo que no fueron aceptados y el motivo del porque no fueron aceptados

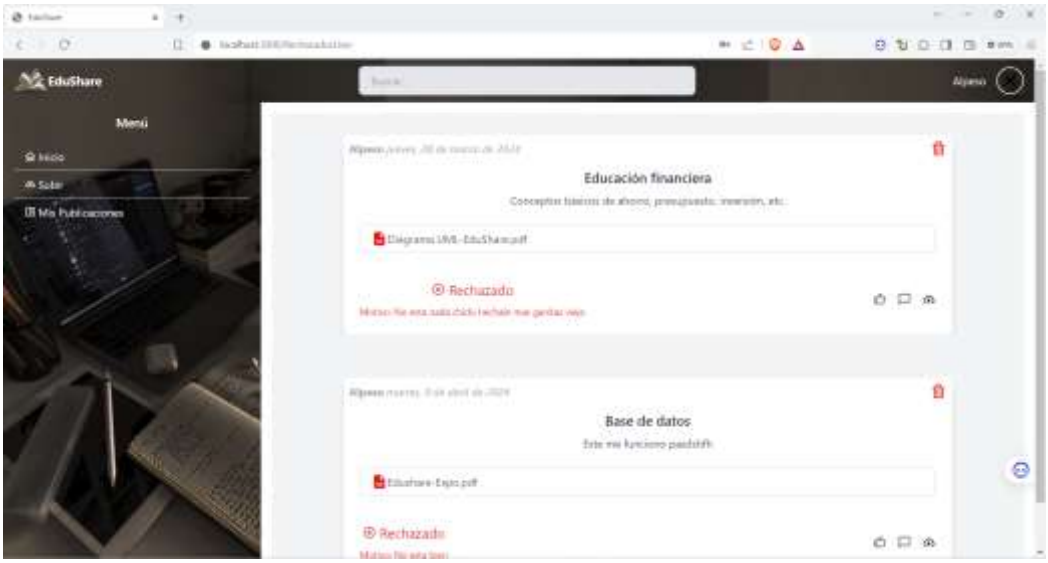


Imagen 27

Perfil

Esto es para ver la información de los usuarios, y tomar una mejor decisión cuando se valla o no, aceptar una publicación de ese usuario, e igual al revisar el tuyo puedes editarlo en caso de que sea tuyo y los perfiles de los demás solo puedes ver la informacion

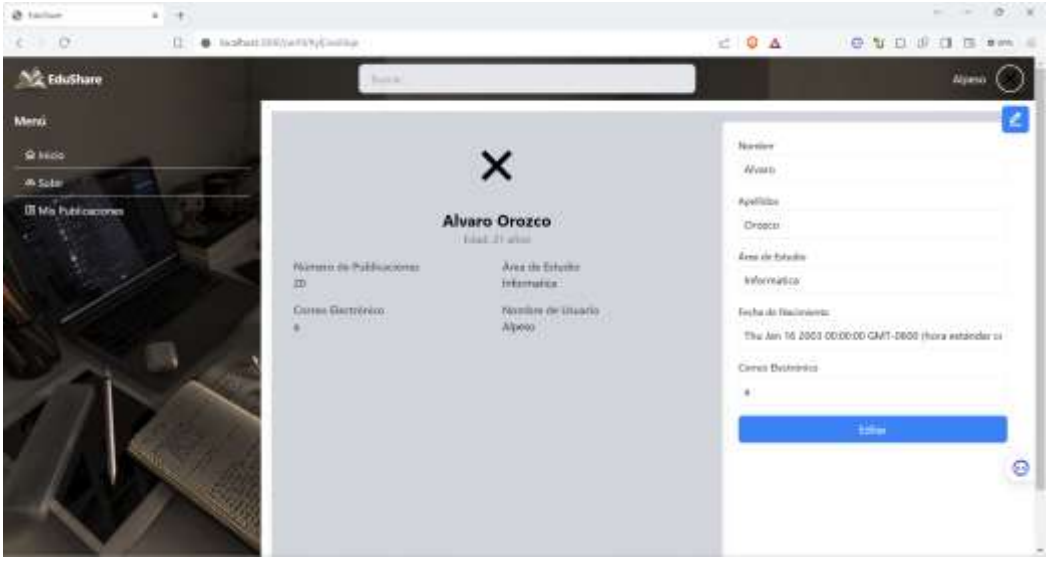


Imagen 28

Comentarios

En esta parte tiene muchos accesos para ir al perfil, insert para agregar los comentarios u un acceso directo para mandar mensajes

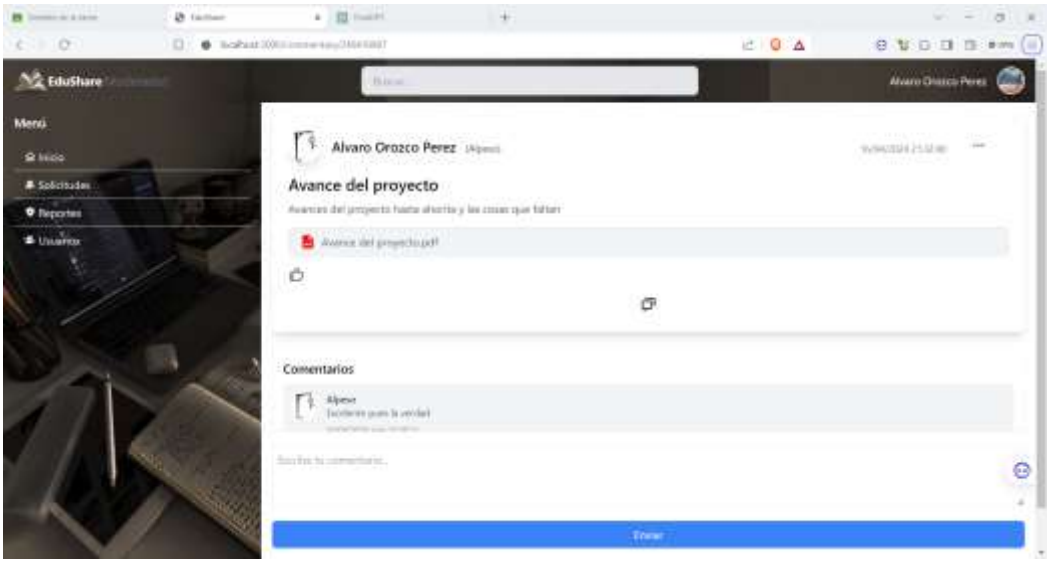


Imagen 29

Reporte

Esta dentro de comentarios puedes reportar cualquier contenido y el moderador sera el responsable de verificar que la información deberá estar en la plataforma

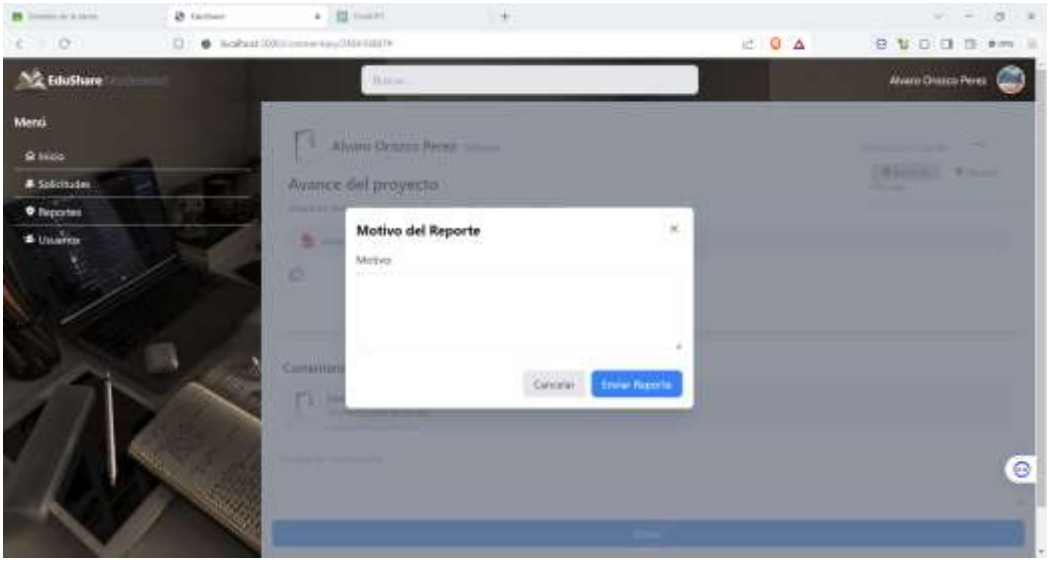


Imagen 30

Solicitudes Pendientes

En esta parte cada que un usuario quiera compartir su contenido con la comunidad, el moderador podrá ver las peticiones de una forma mas sencilla, para después revisarla una por una

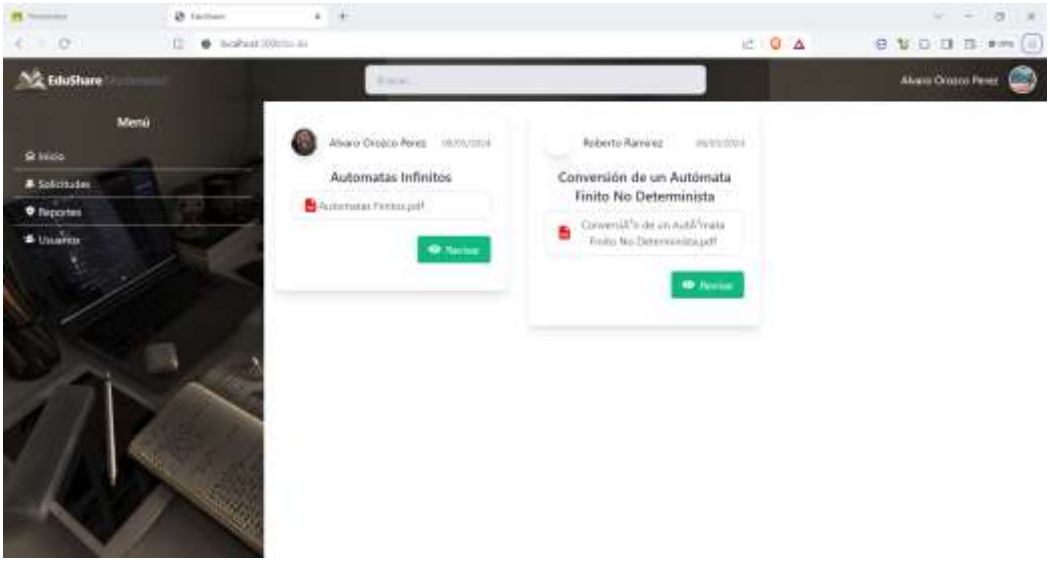


Imagen 31

Revisión

En esta parte es donde el moderador revisa la información de la publicación, revisar el perfil y abrir el archivo desde el navegador

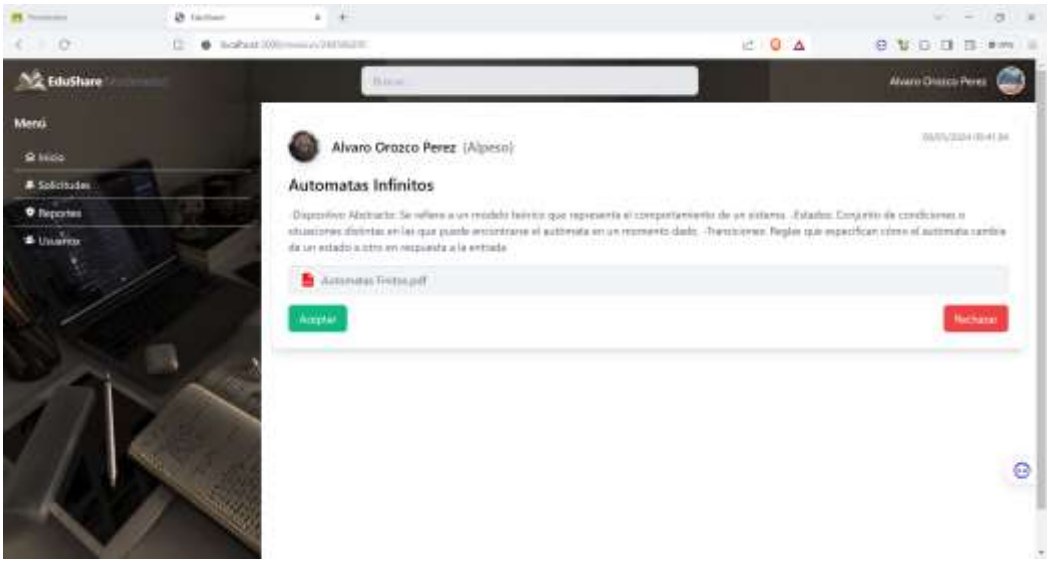


Imagen 32

Rechazo (Formulario)

Esta parte es para dar el motivo del rechazo, un pequeño menú que se despliega al darle clic

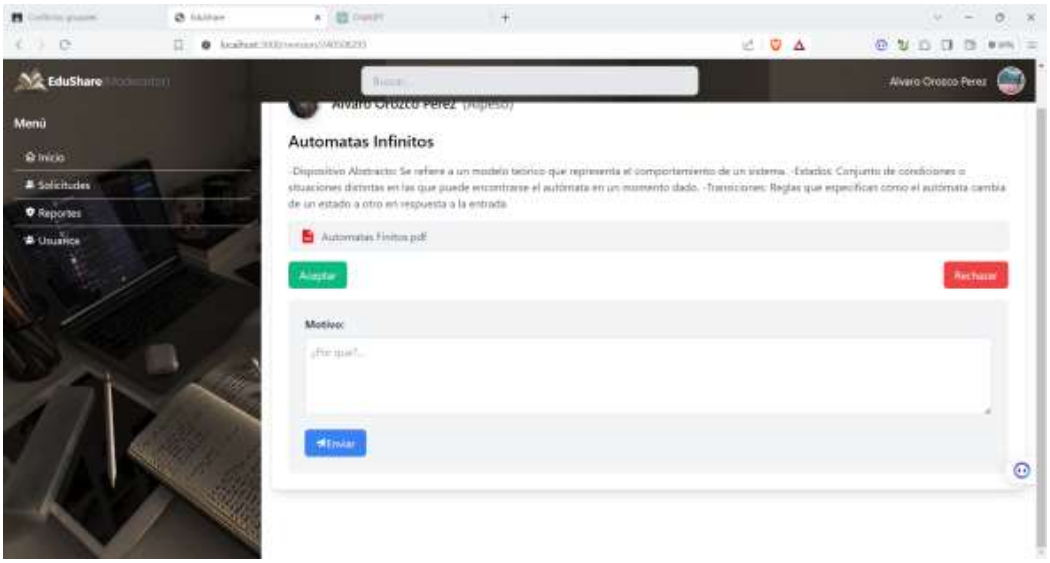


Imagen 33

Rechazados

En esta parte es para lo que ya reviso el moderador pero no fueron aptos para la plataforma o incumplen alguna norma

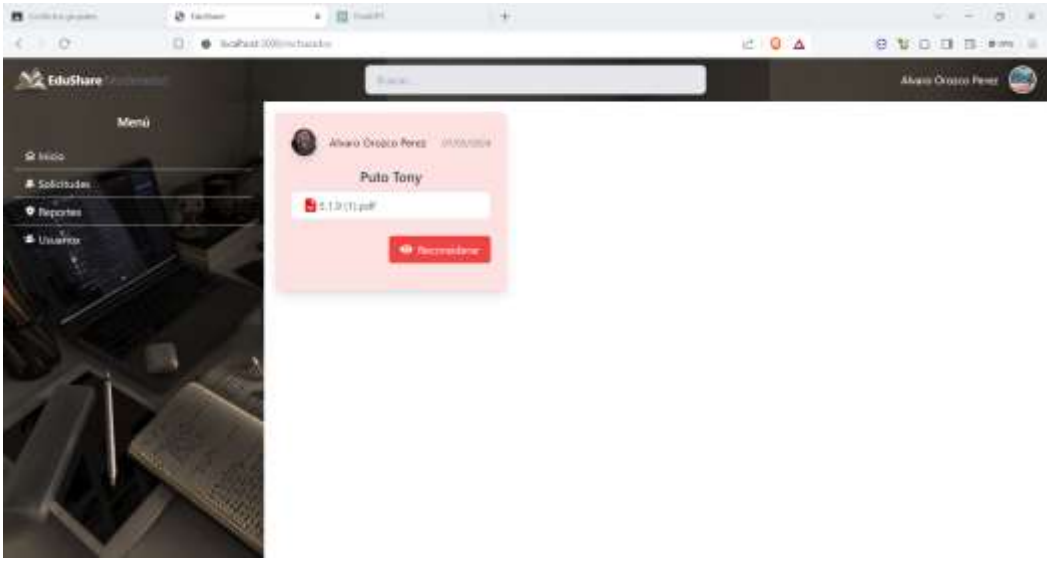


Imagen 34

Reportes

En esta revisión no cambia mucho a la anterior pero muestra los reportes sobre

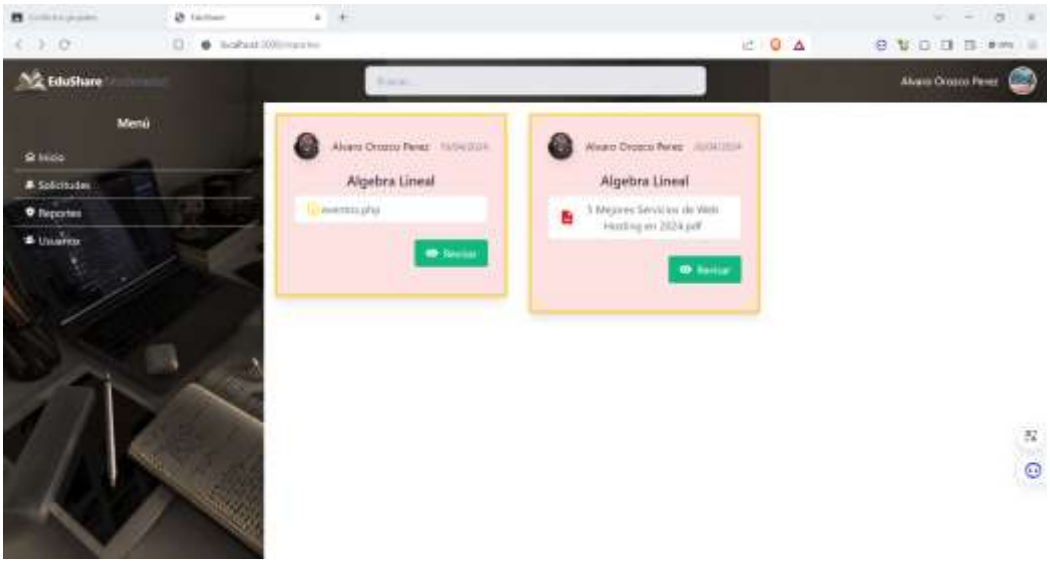


Imagen 35

Revisión de reportes

Aquí están los comentarios o el motivo le reporte que le ponen los usuarios, como e suna plataforma pequeña todavía puedes expresarte libremente

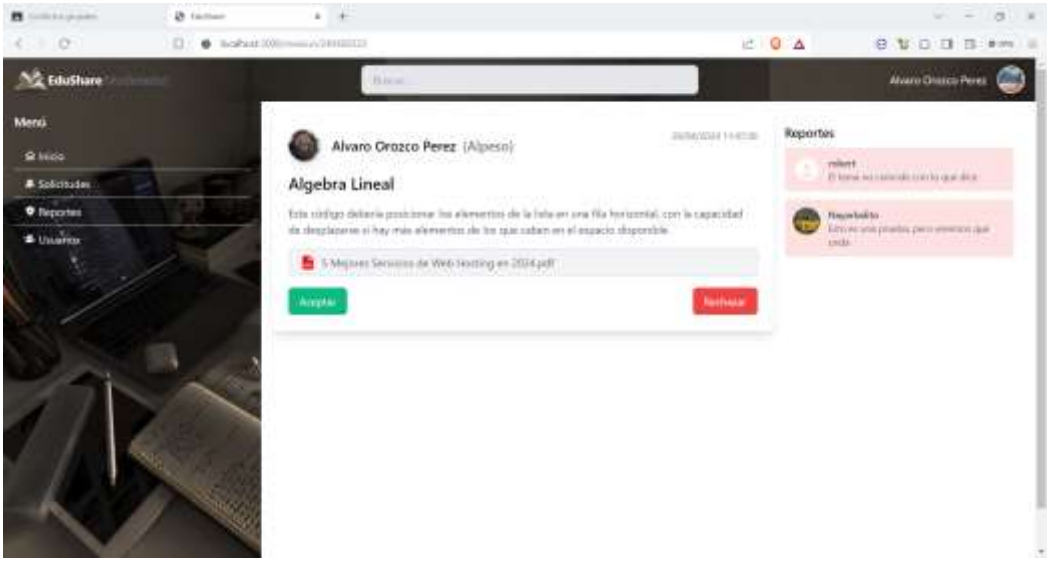


Imagen 36

Rechazo (Formulario)

Este pequeño menú desplegable el cual se puede quitar y poner con un boton

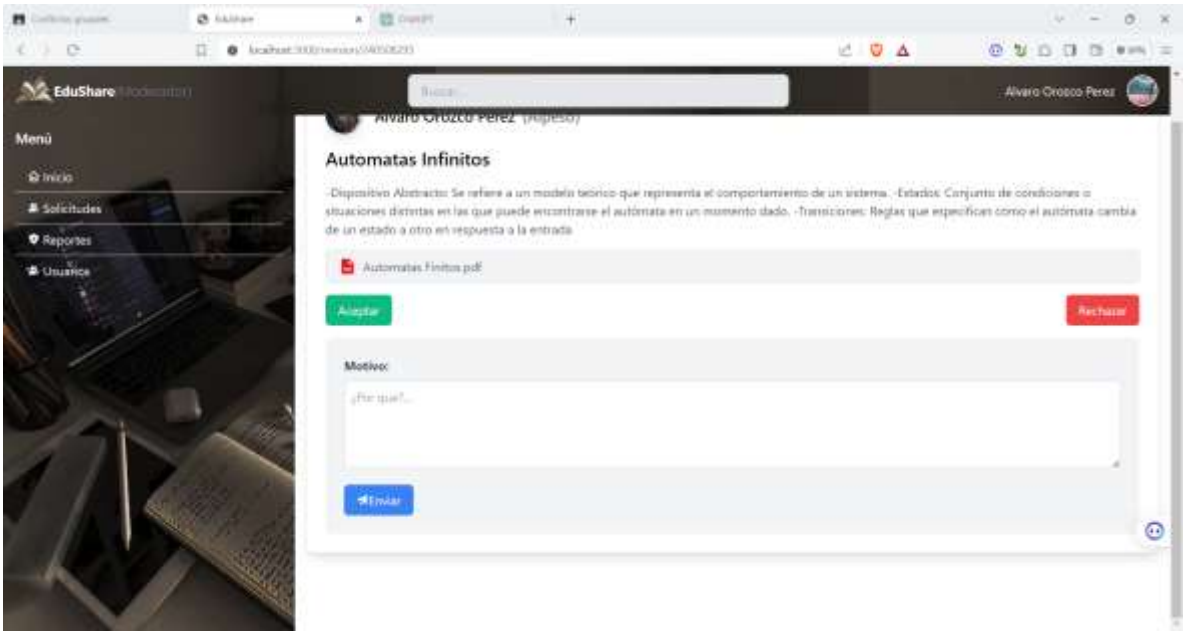


Imagen 37

Control de usuarios

Esta parte es funcional para la administración de los usuarios, la capacidad de bloquearlos o de reescribir informacion

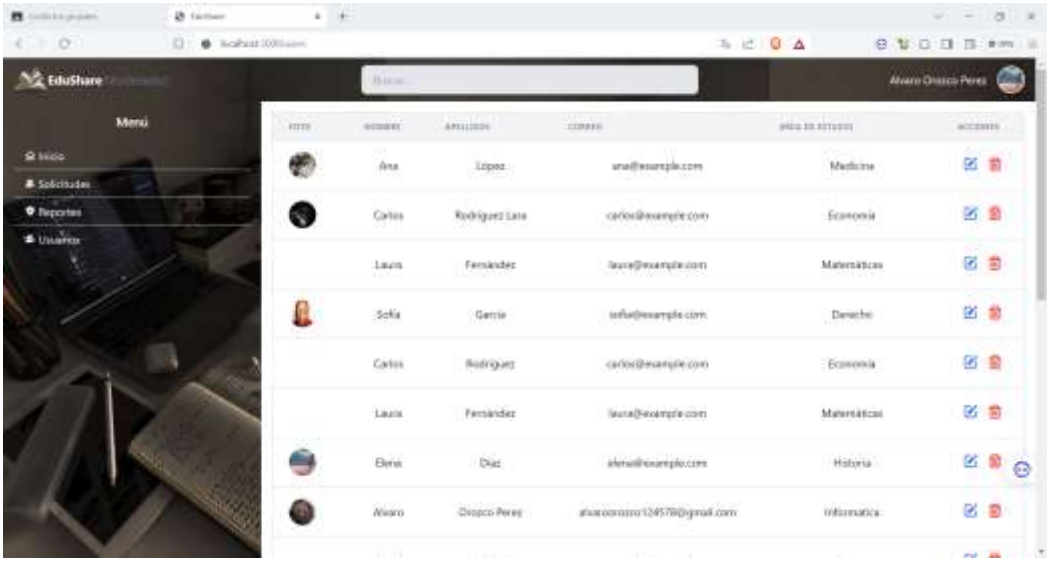


Imagen 38

Login

Codigo: Aquí pues esta todo el diseño de la ventana, el formulario y todo el frond end, después esta el controlador, que a grandes rasgos es la consulta

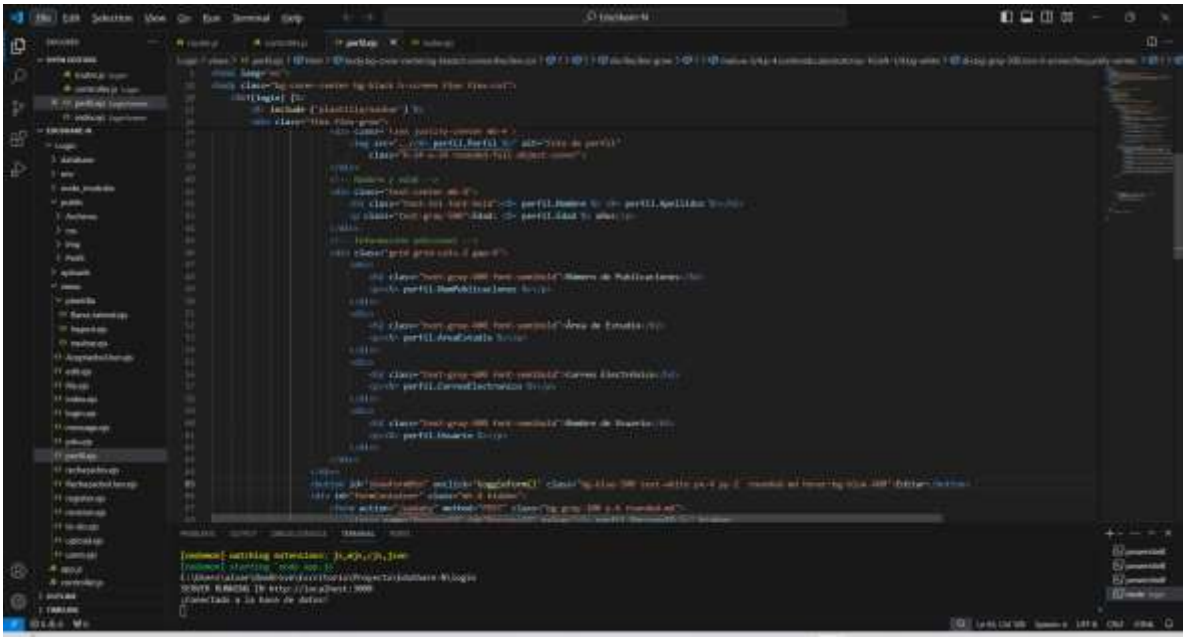


Imagen 39

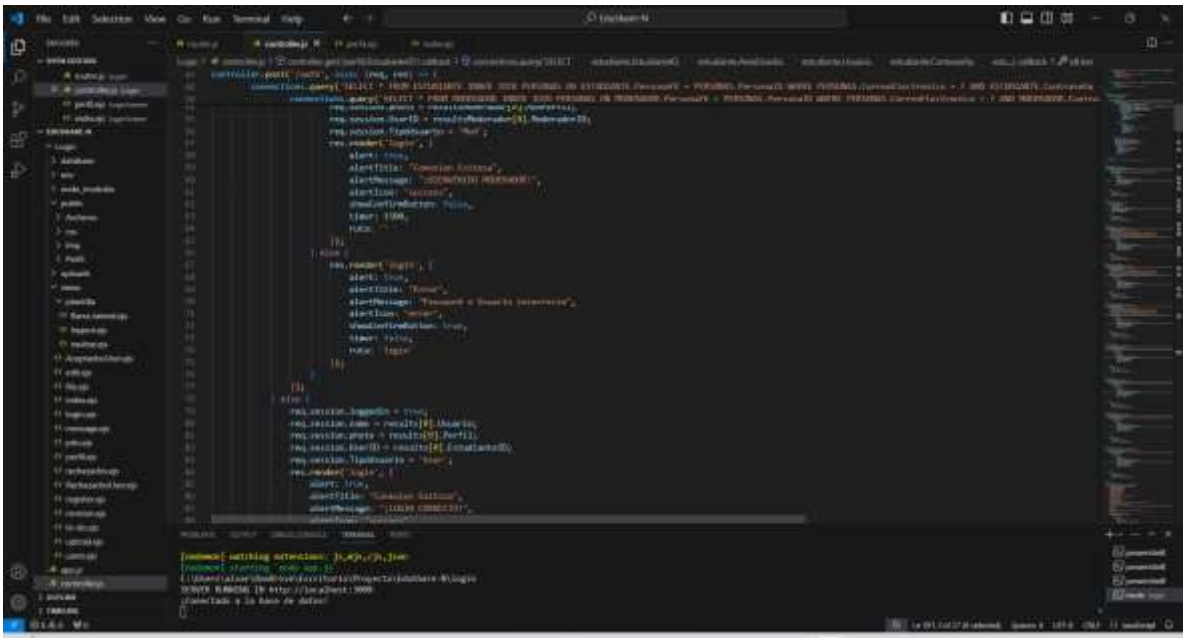


Imagen 40

Registro

Código: Aquí está el formulario como tal el código, para que visualmente se vea mejor, después está el controlador que este como tal es el INSERT para insertar los datos en la base de datos de manera permanente

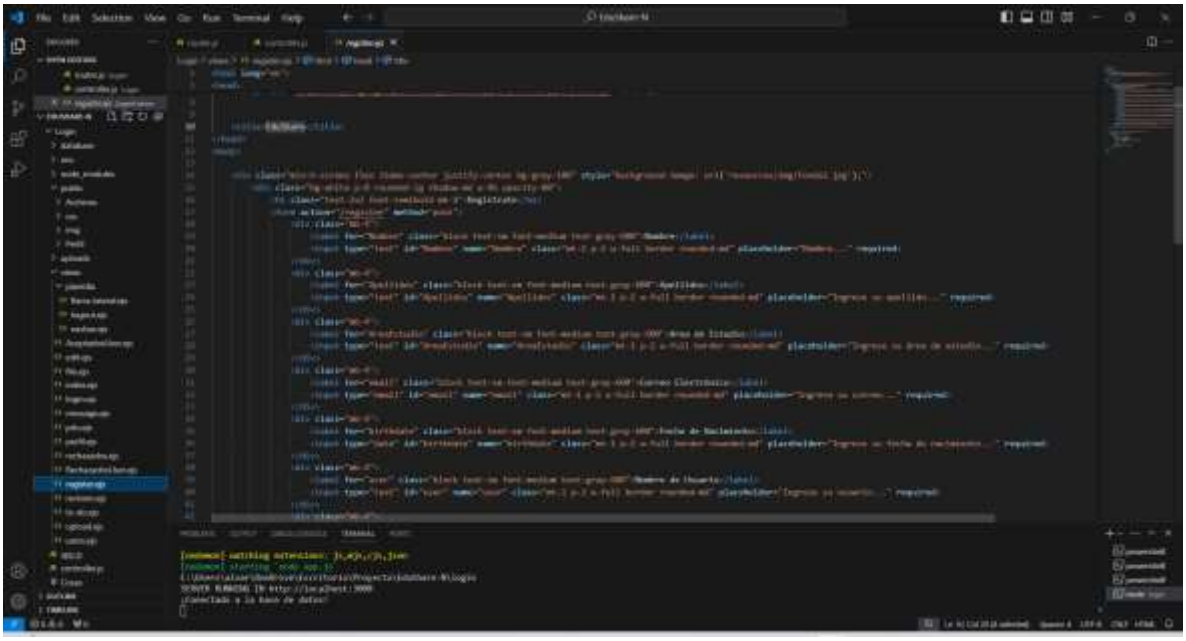


Imagen 41

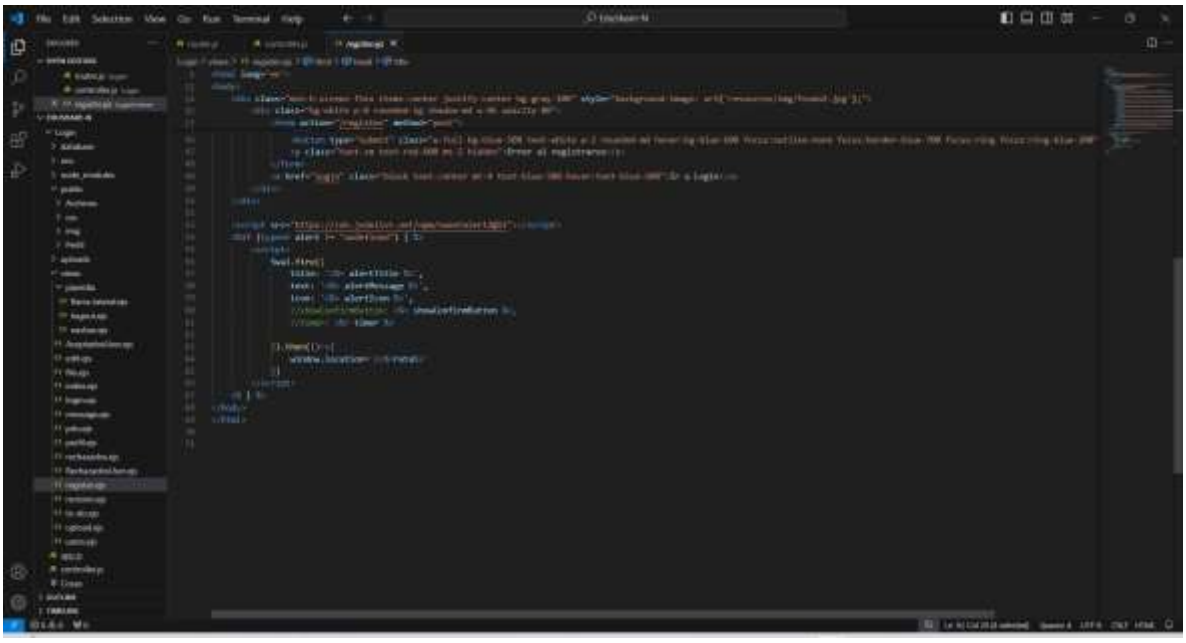


Imagen 42

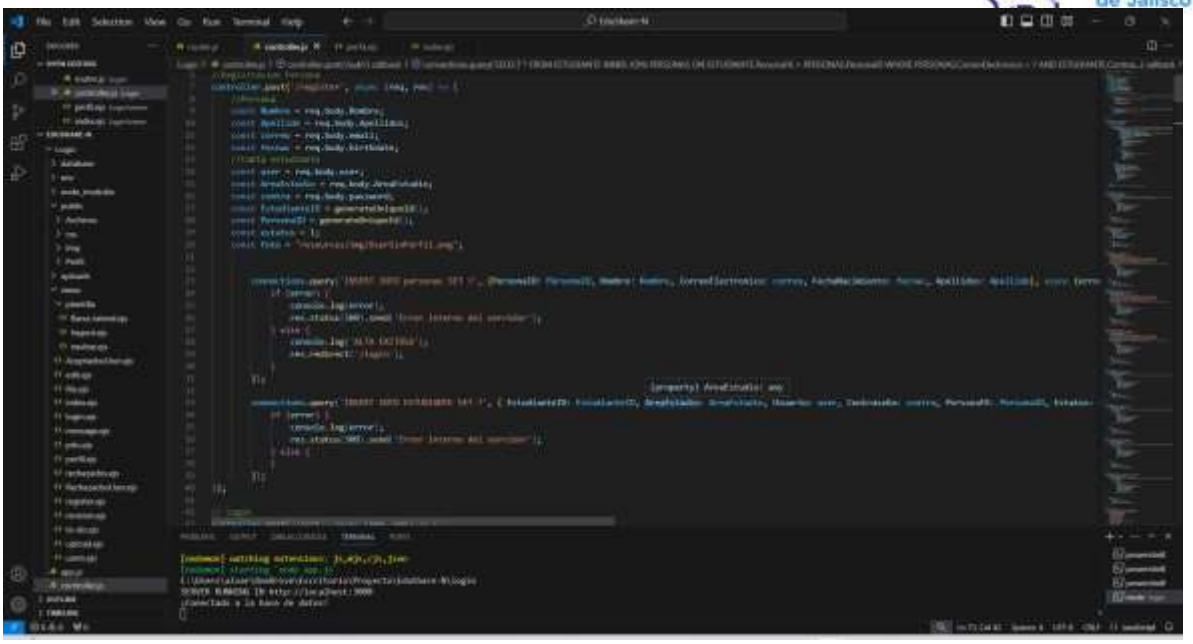


Imagen 43

Inicio

Código: Aquí está todo en código que básicamente esta es una etiqueta selection esta en un bucle determinada por el número de respuestas que tenga la base de datos, lo demás es diseño. En el controlador es esta la consulta en la base de datos y guardada en arreglo para mostrar la información

Imagen 44

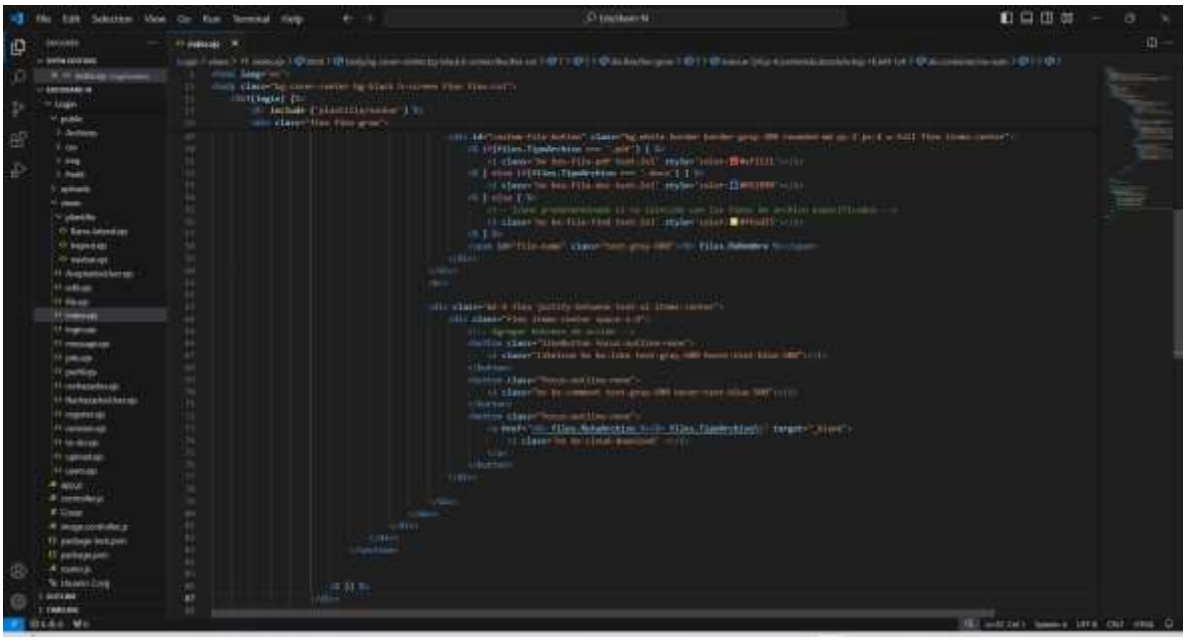


Imagen 45

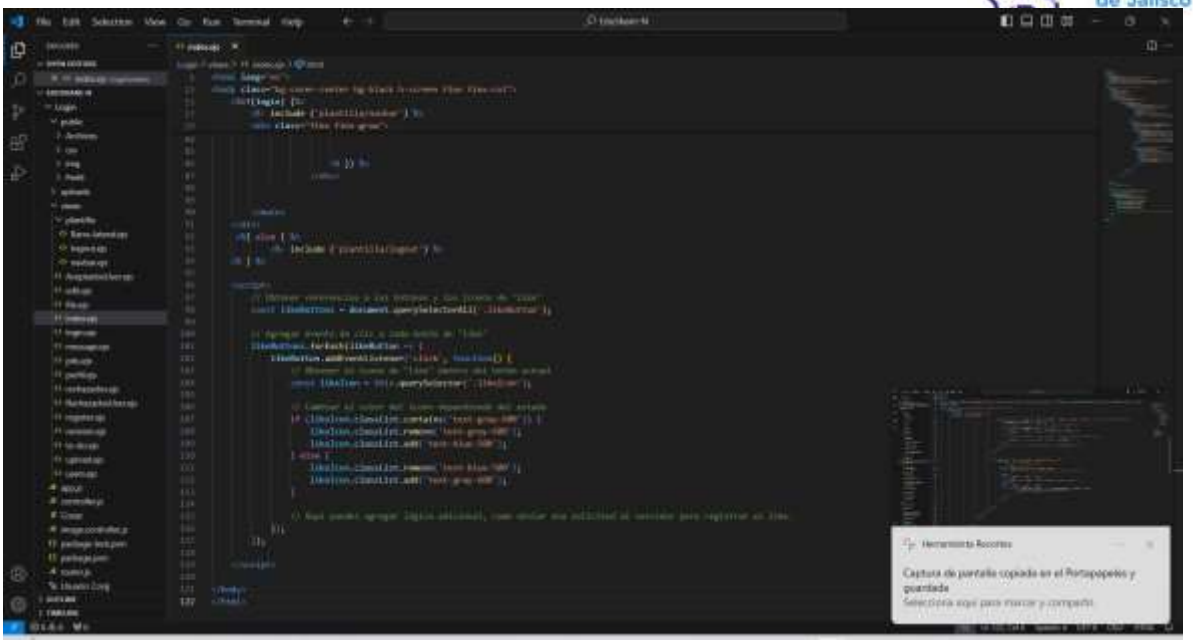


Imagen 46

Subir archivos

Codigo: En esta parte esta el insert, pero también utilice la librería “multer” para subir toda clase de archivos y que fuera compatible con ellos, lógica es muy sencilla sin embarlo la librería es algo tediosa de utilizar

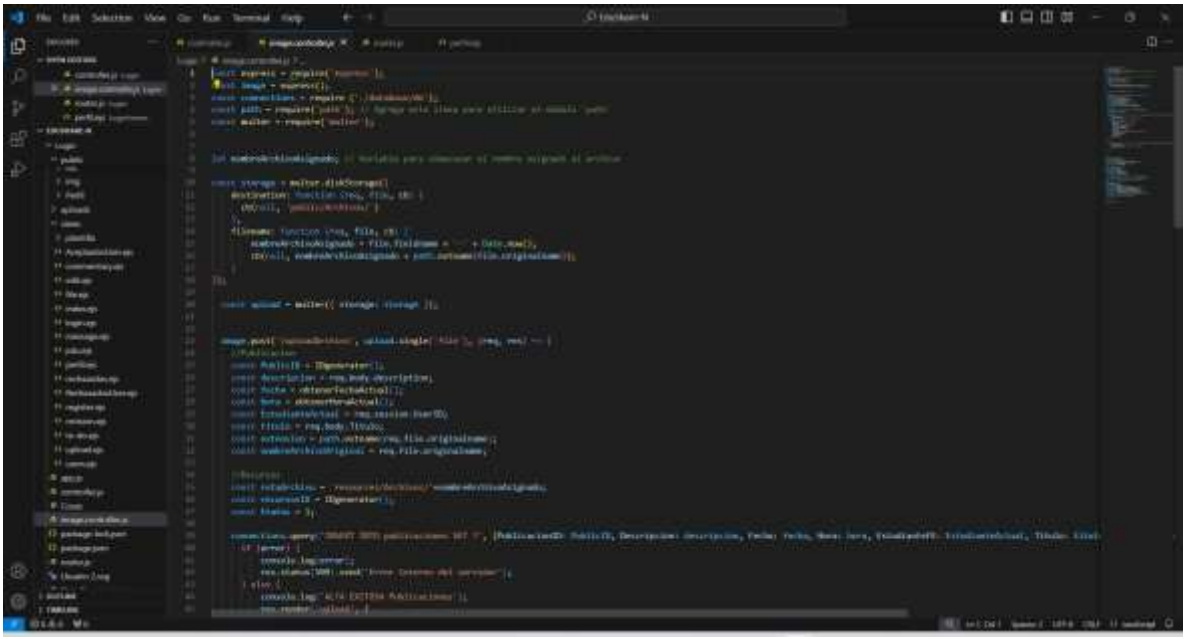


Imagen 47

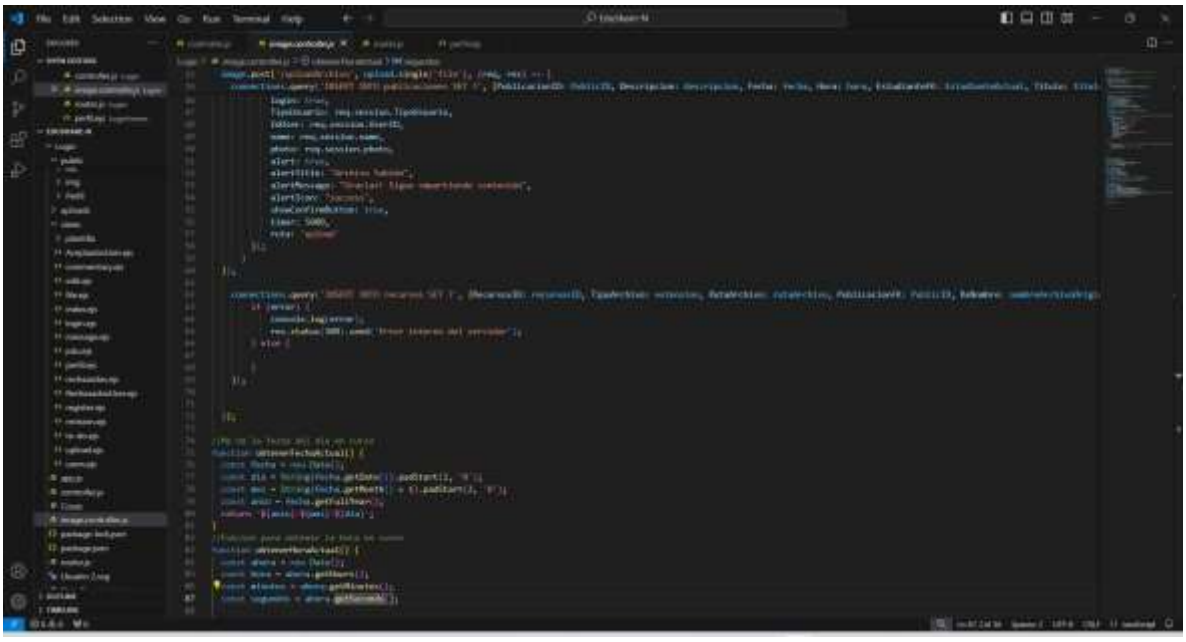


Imagen 48

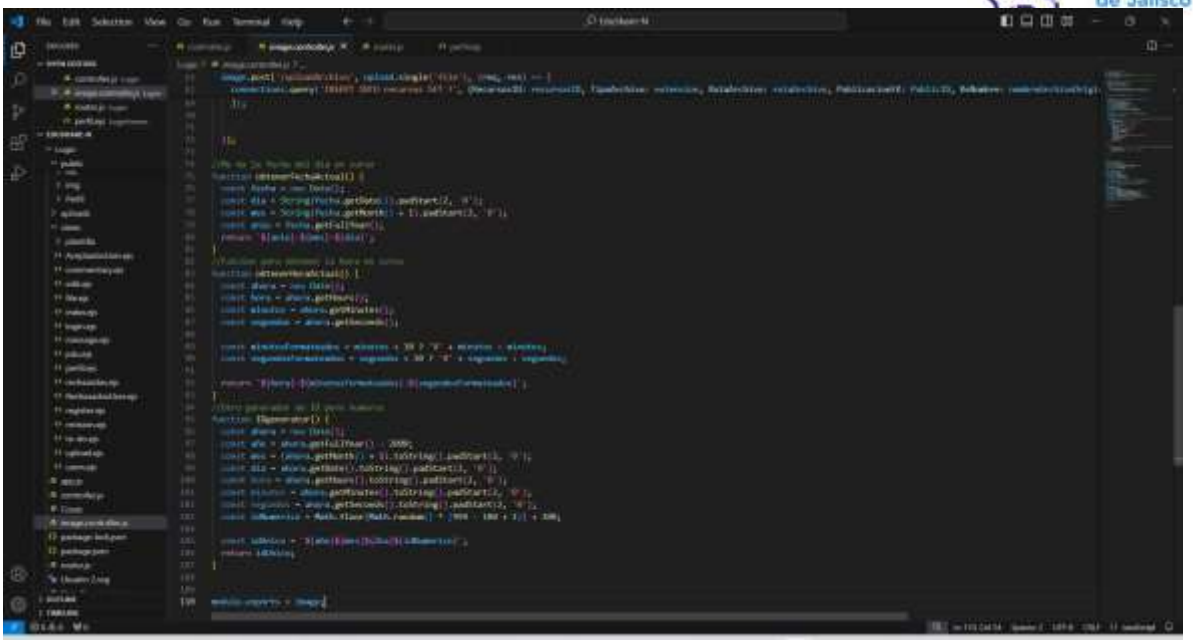


Imagen 49

Mis Publicaciones

Código: esta es la consulta de vista y los Updates correspondientes a la eliminación lógica y al edit

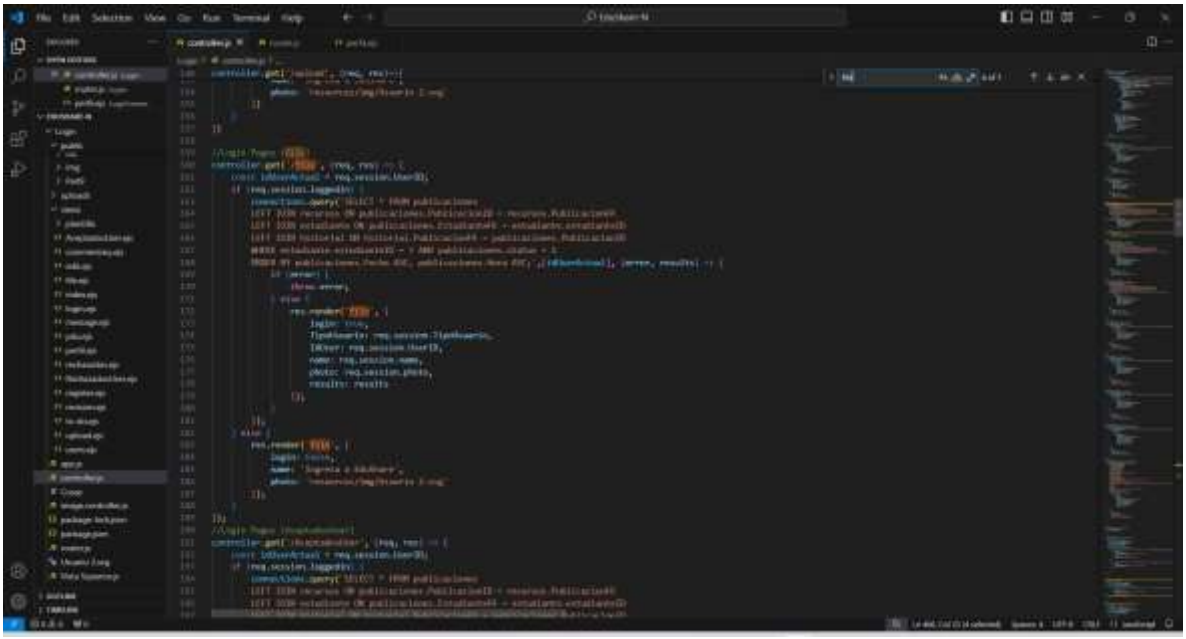


Imagen 50

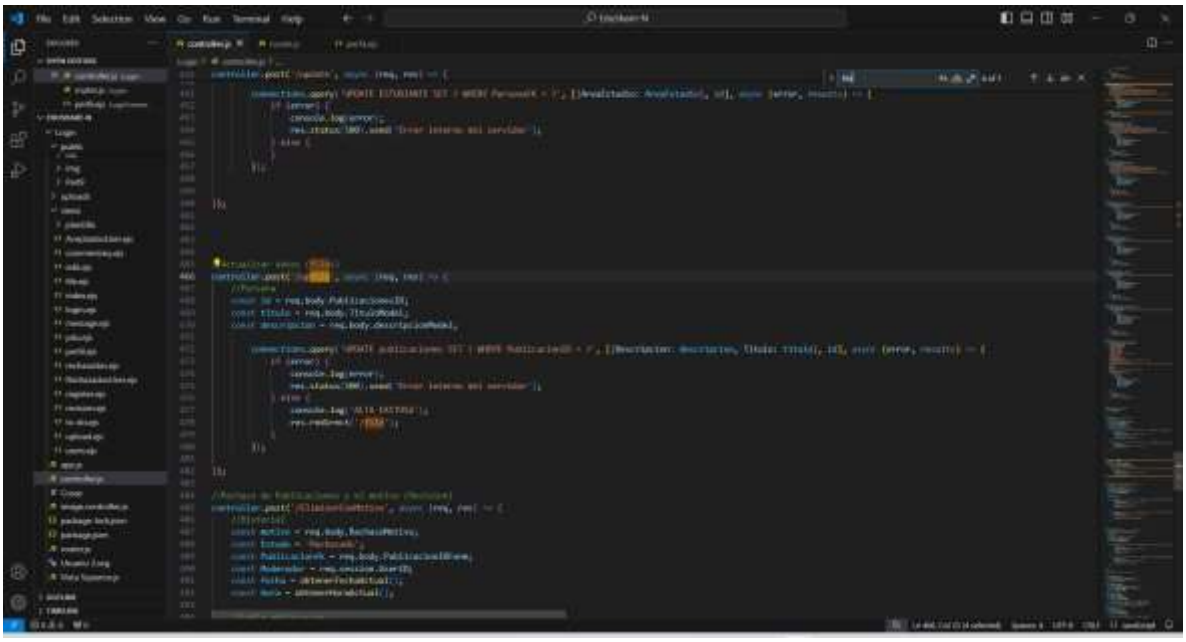


Imagen 51

```
//Elimina y redirige a /file
router.get('/defile/:PublicacionID', (req, res)=>{
    const id = req.params.PublicacionID;
    connections.query('UPDATE publicaciones SET status = 4 WHERE PublicacionID = ?', [id], (error, results)=>{
        if(error){
            throw error;
        } else {
            res.redirect('/file');
        }
    })
})
```

Imagen 52

Aceptados

Código: Aquí se encuentra el Front End de la vista

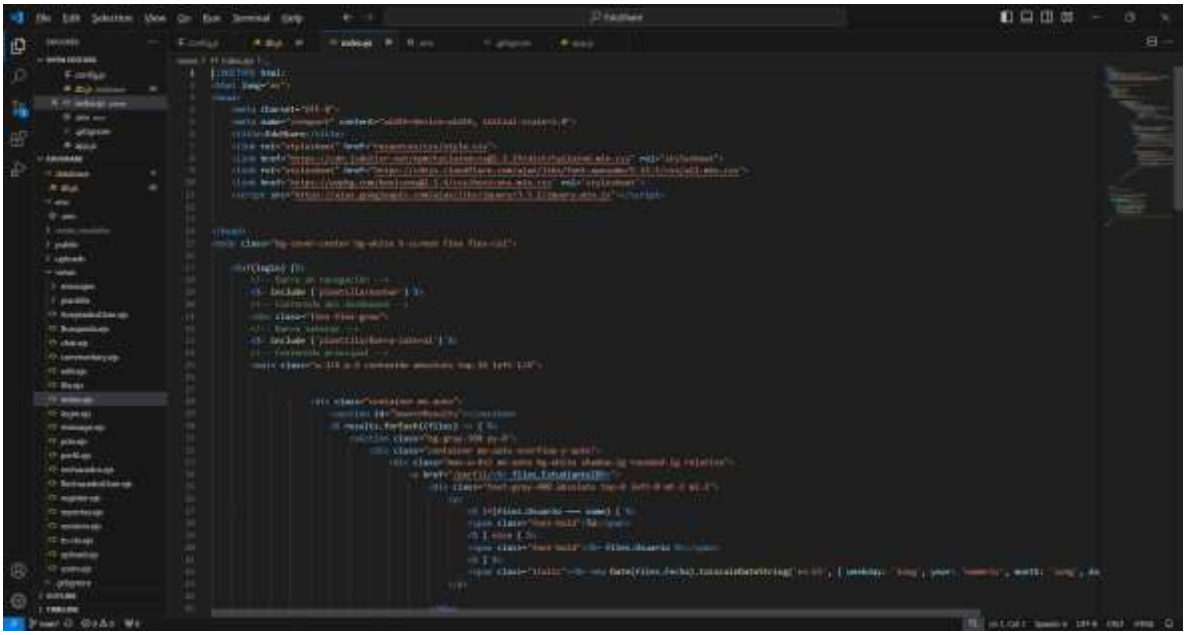


Imagen 53

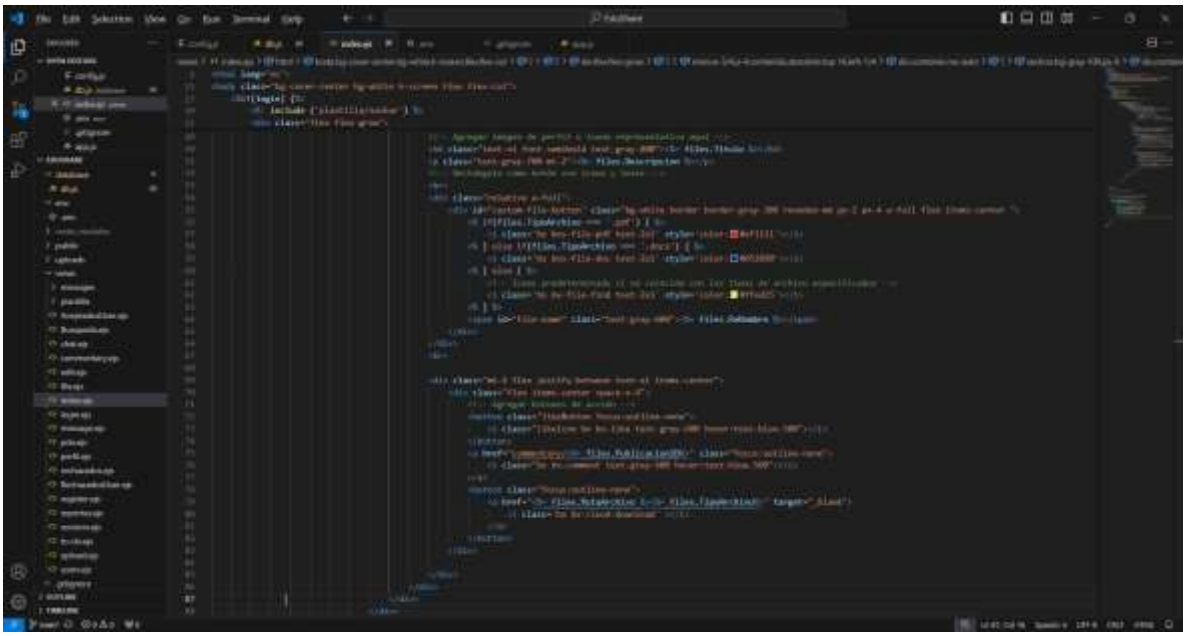


Imagen 54

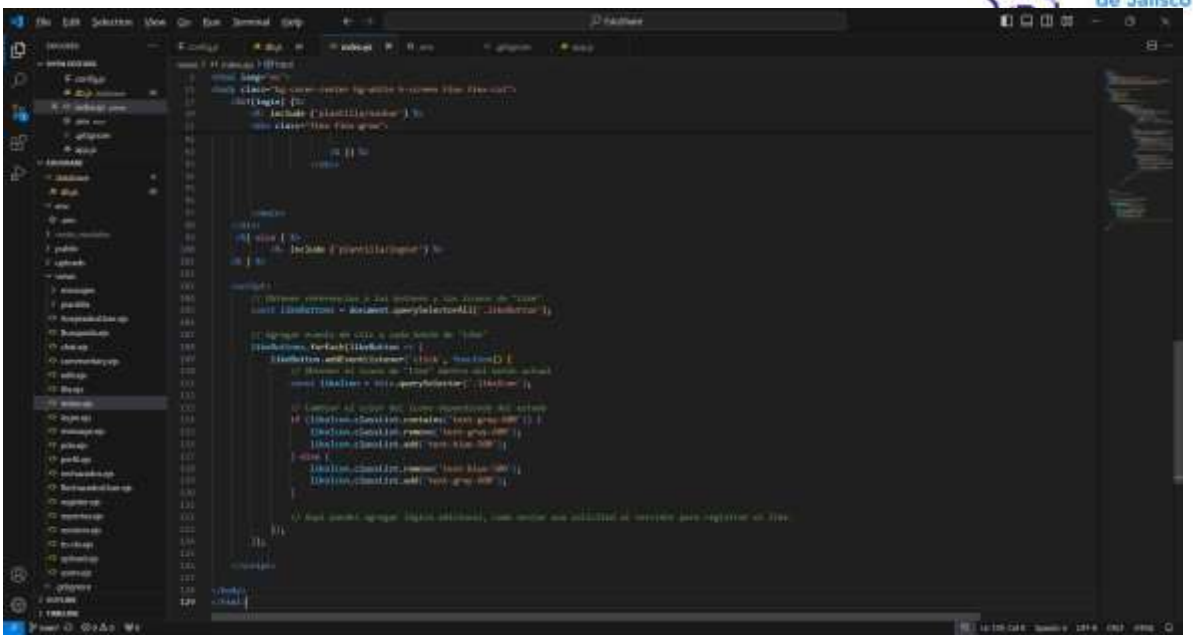


Imagen 55

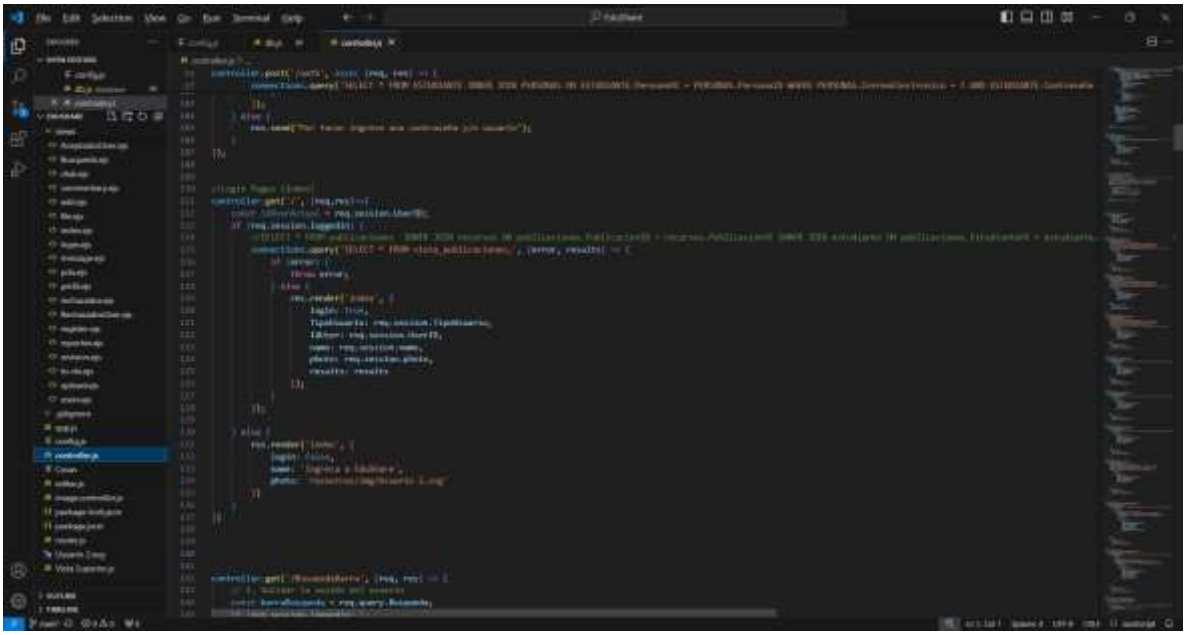


Imagen 56

Rechazados

Código: esta es la consulta de vista y los Updates correspondientes a la eliminación lógica y al edit

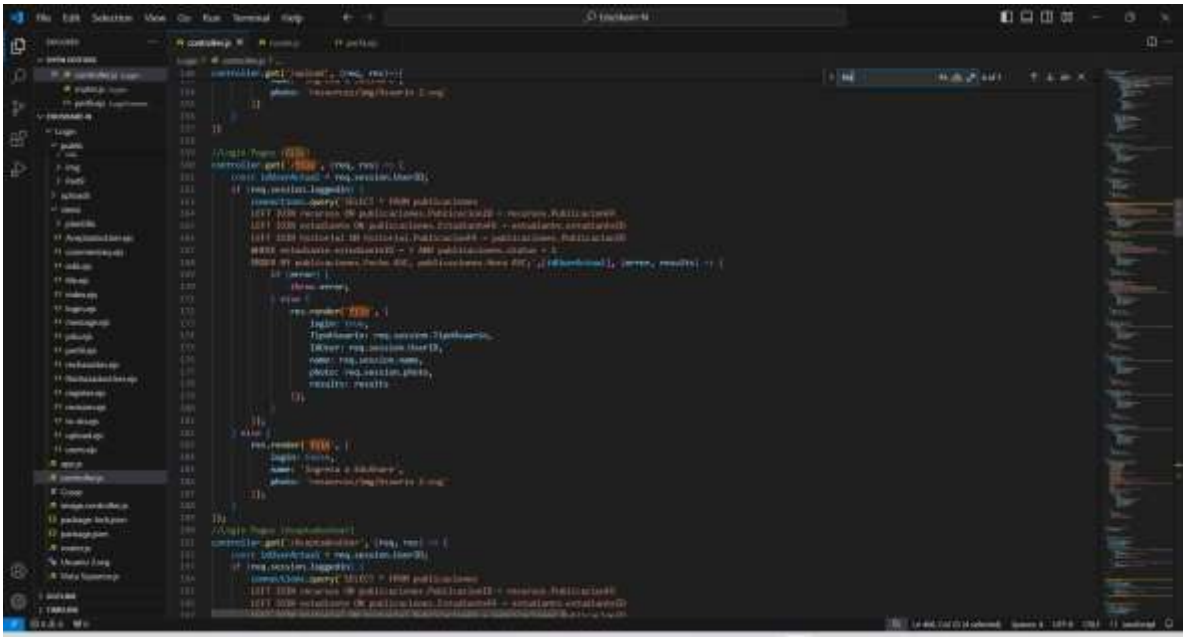


Imagen 57

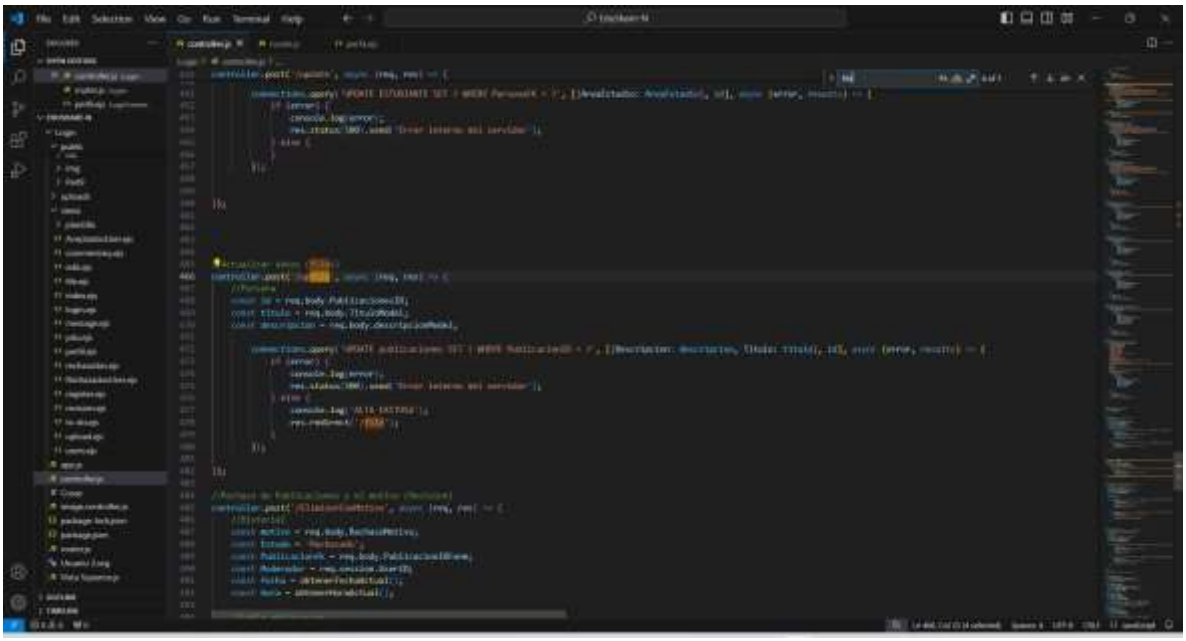


Imagen 58

```
//Elimina y redirige a /file
router.get('/defileRechazados/:PublicacionID', (req, res)=>{
    const id = req.params.PublicacionID;
    connections.query('UPDATE publicaciones SET status = 4 WHERE PublicacionID = ?', [id], (error, results)=>{
        if(error){
            throw error;
        } else {
            res.redirect('/AceptadosUser');
        }
    })
})
```

Imagen 59

Perfil

Código: Esta es solo la consulta en la base de datos que ayuda a que todo funcione correctamente

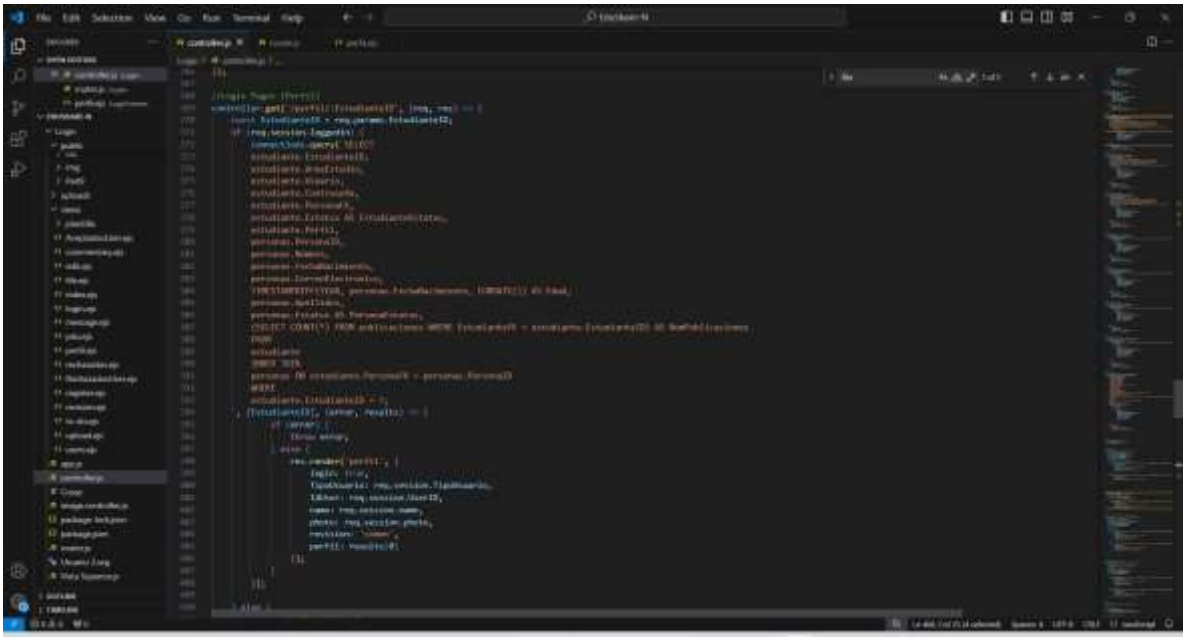


Imagen 60

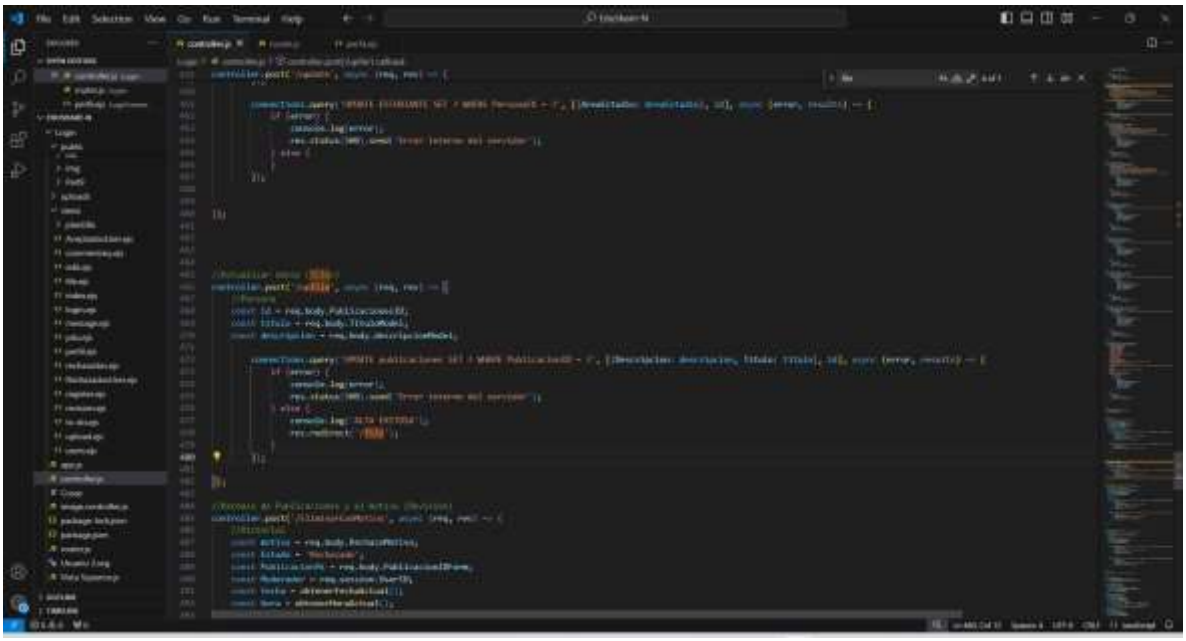


Imagen 61

Edición de mi perfil

En esta parte puedes cambiar mas cosas de su perfil, como la foto y datos generales

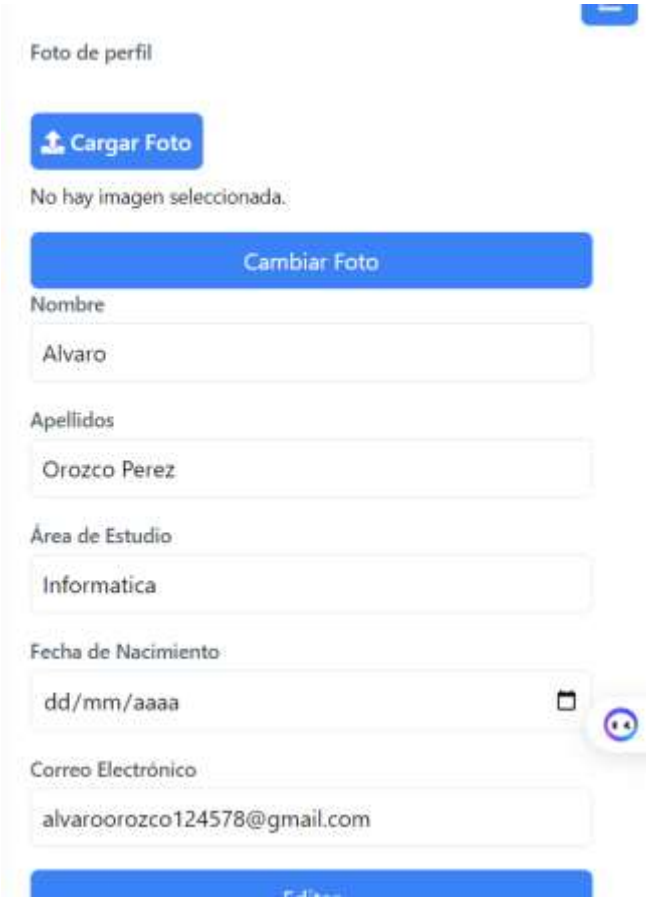


Foto de perfil
Cargar Foto
No hay imagen seleccionada.
Cambiar Foto
Nombre
Alvaro
Apellidos
Orozco Perez
Área de Estudio
Informatica
Fecha de Nacimiento
dd/mm/aaaa
Correo Electrónico
alvaroorozco124578@gmail.com

Imagen 62

Codigo: esta la parte del diseño que es necesaria para darle la forma a cada input y etiqueta y la parte del Update en la base de datos por parte del controlador

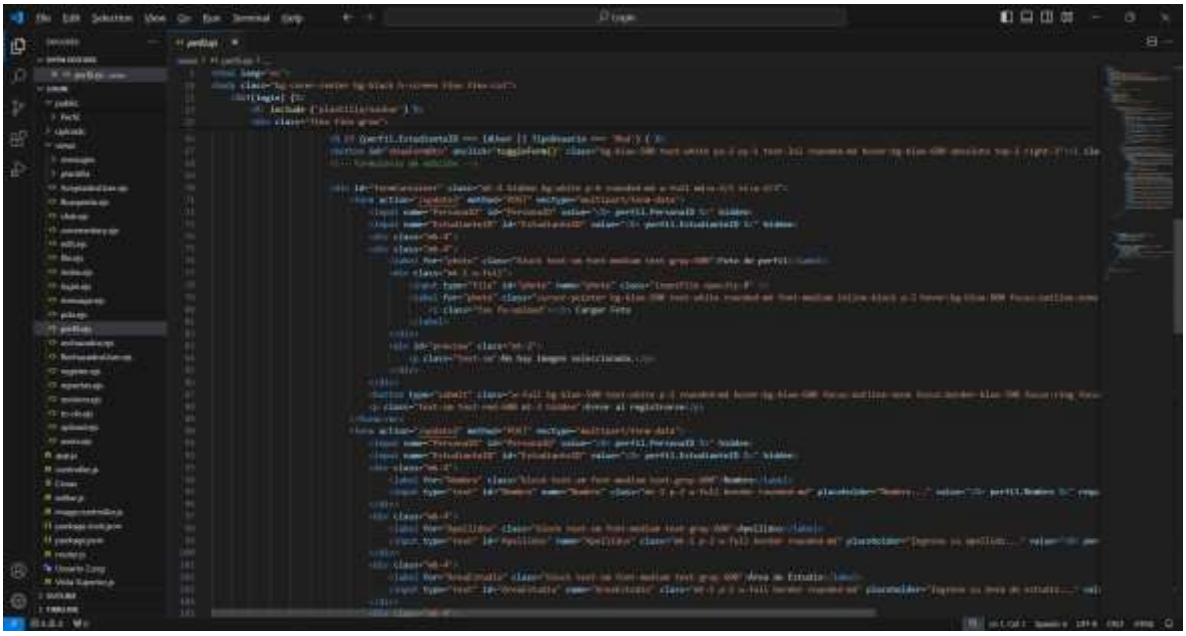


Imagen 63

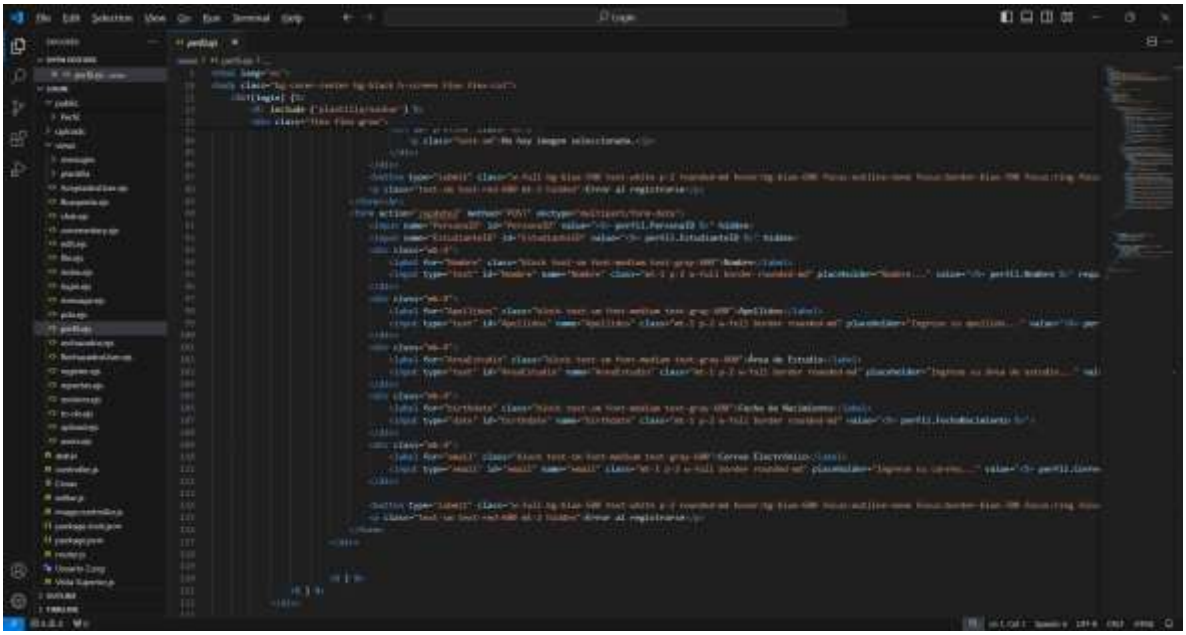


Imagen 64

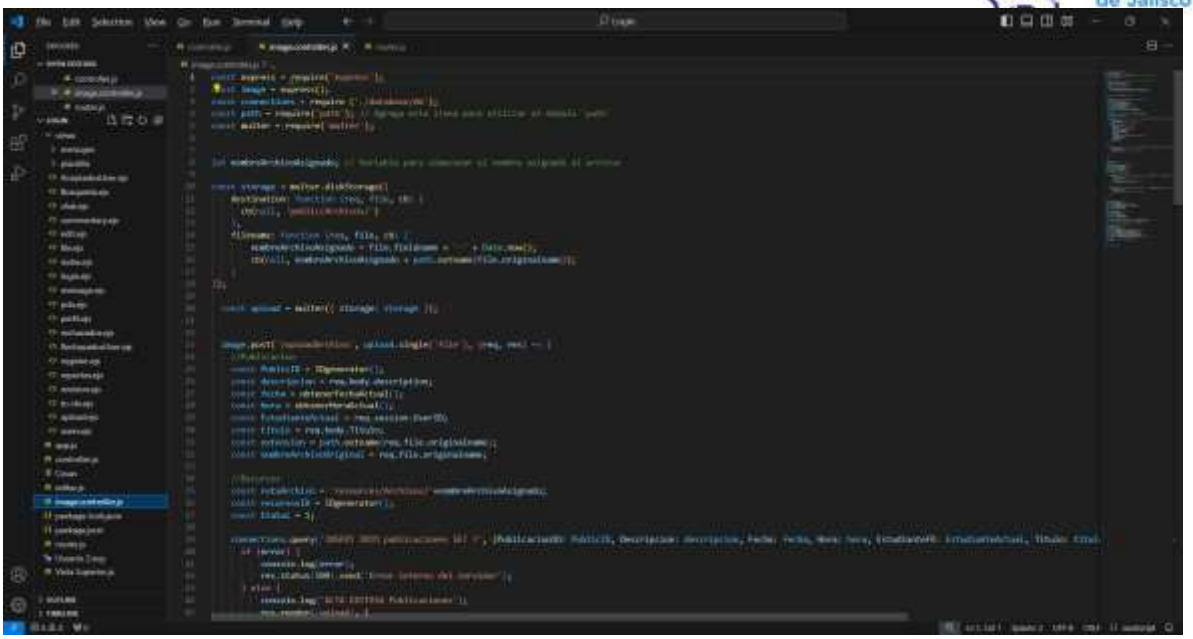


Imagen 65

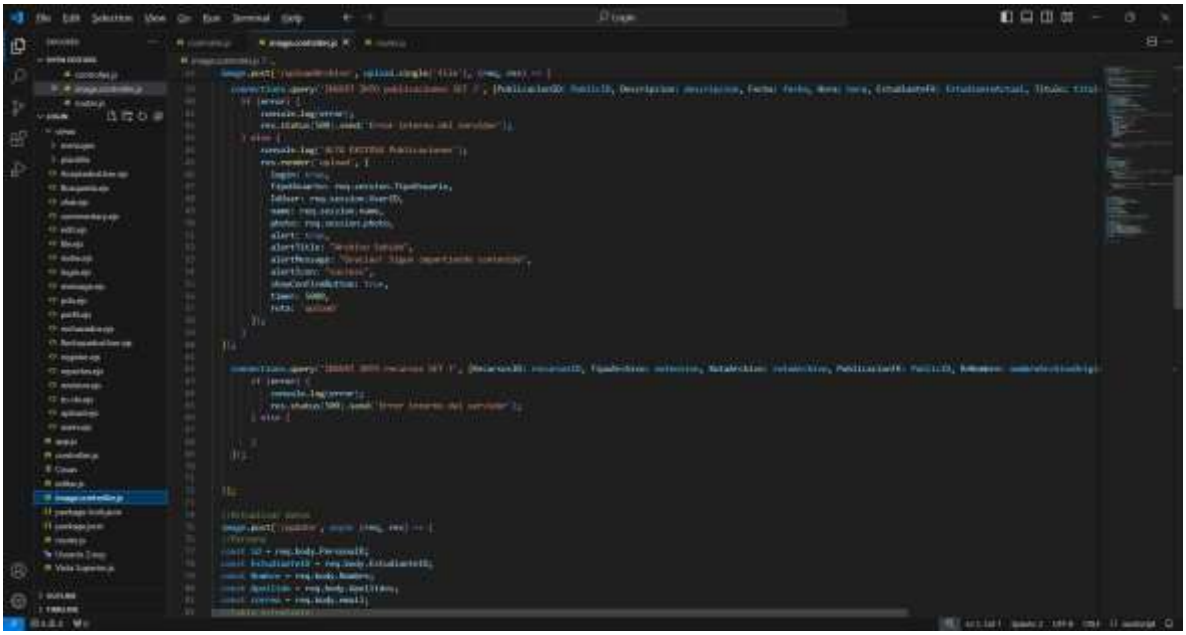


Imagen 66

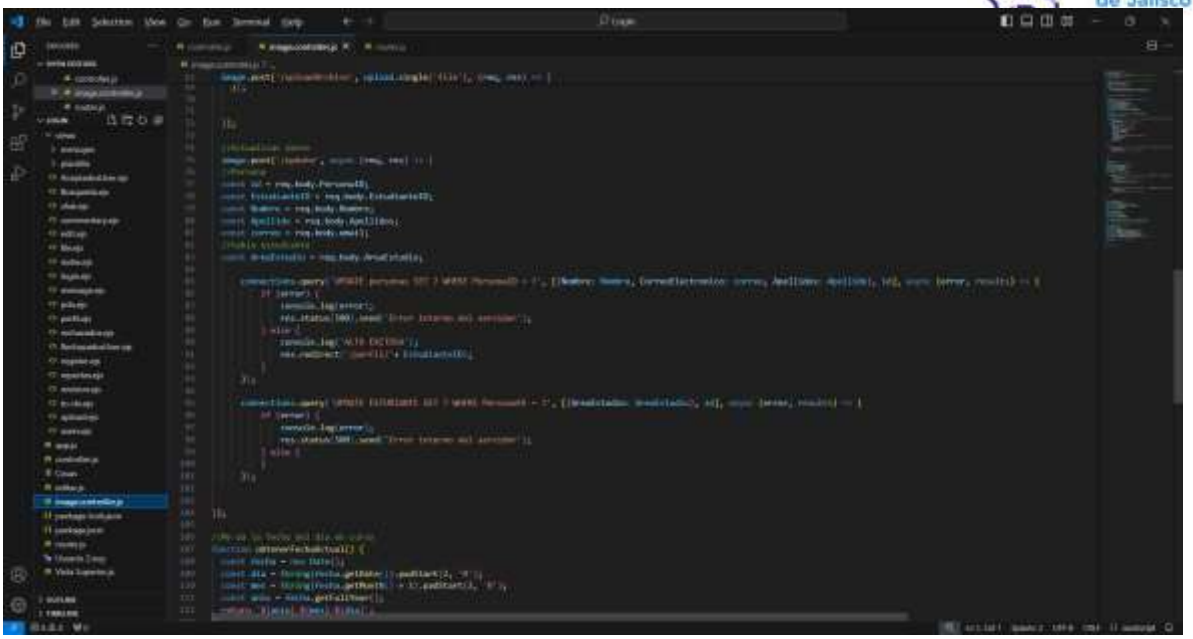


Imagen 67

Comentarios

Código: Aquí esta está el código que hace que el front End y los inserts que se hacen para agregar el comentario a la base de datos y sea funcional para la información

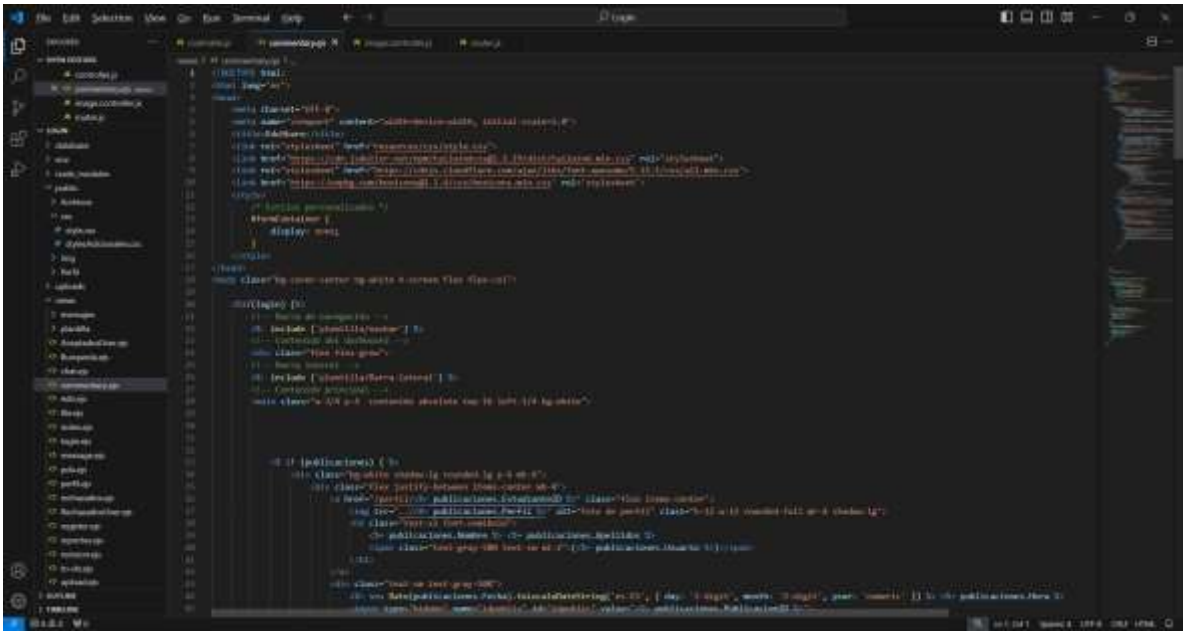


Imagen 68

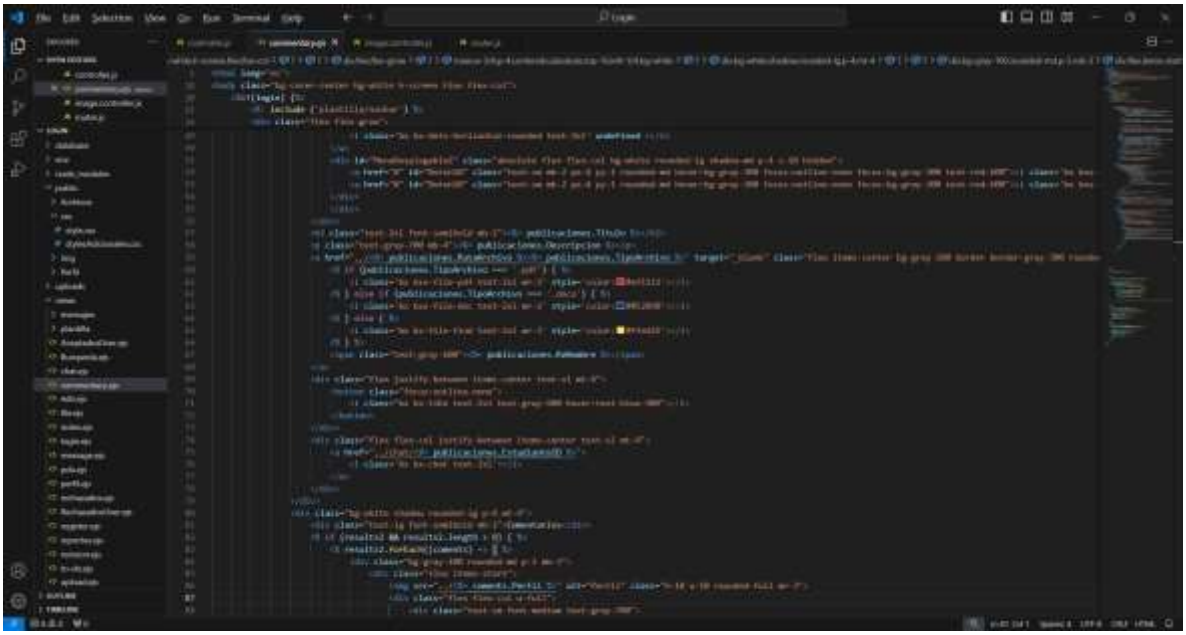


Imagen 69

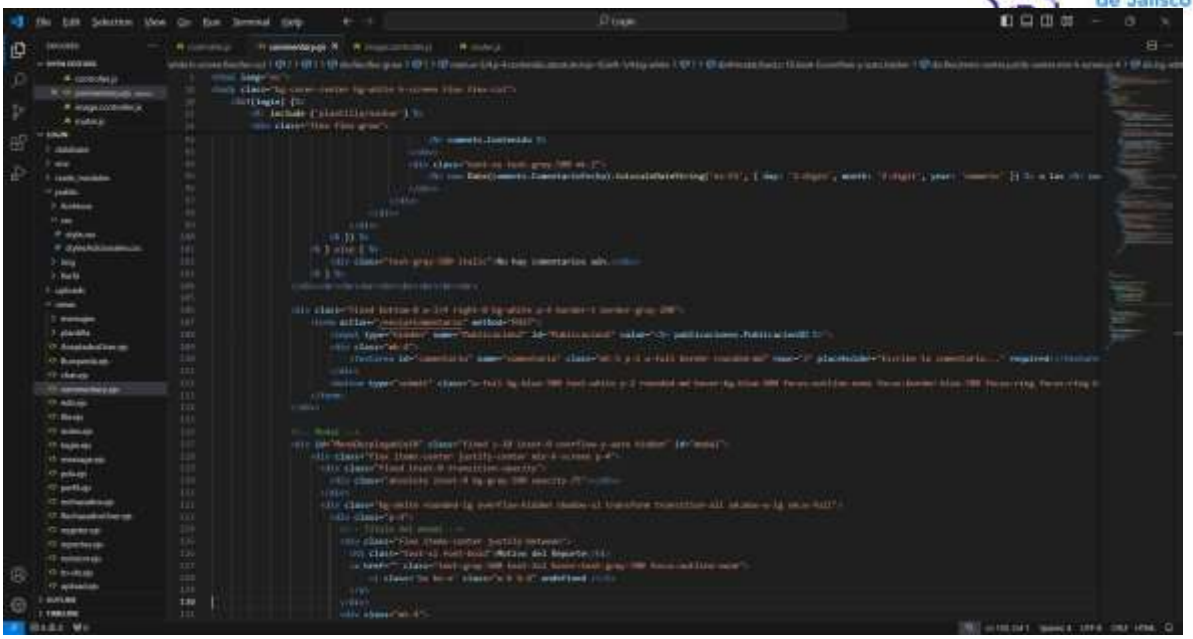


Imagen 70

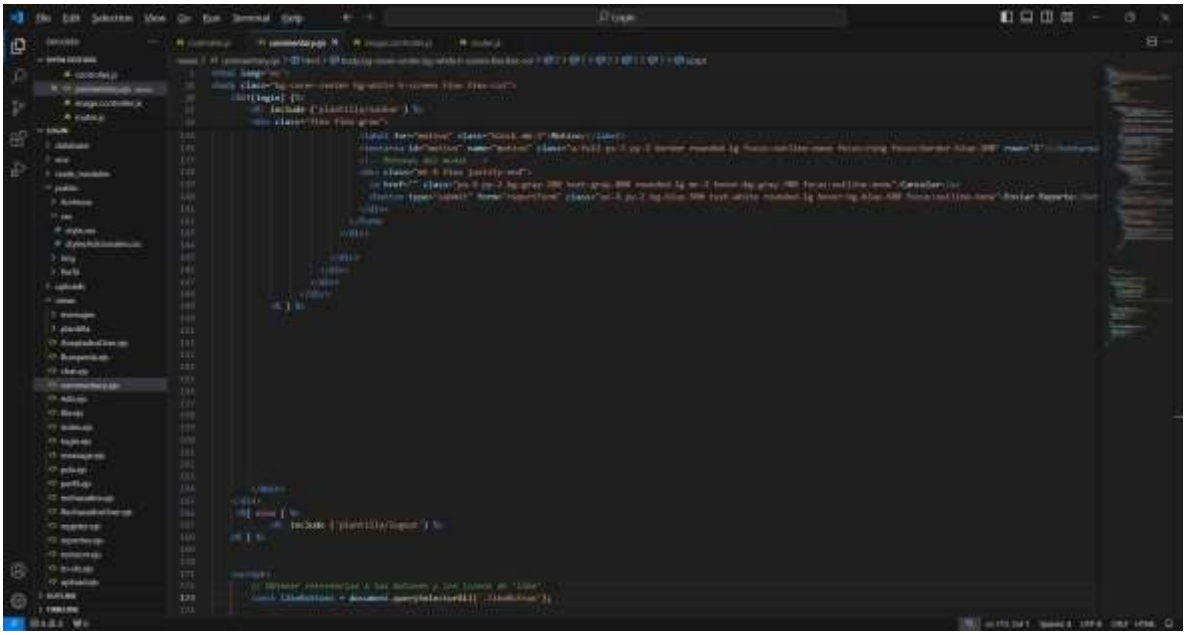


Imagen 71

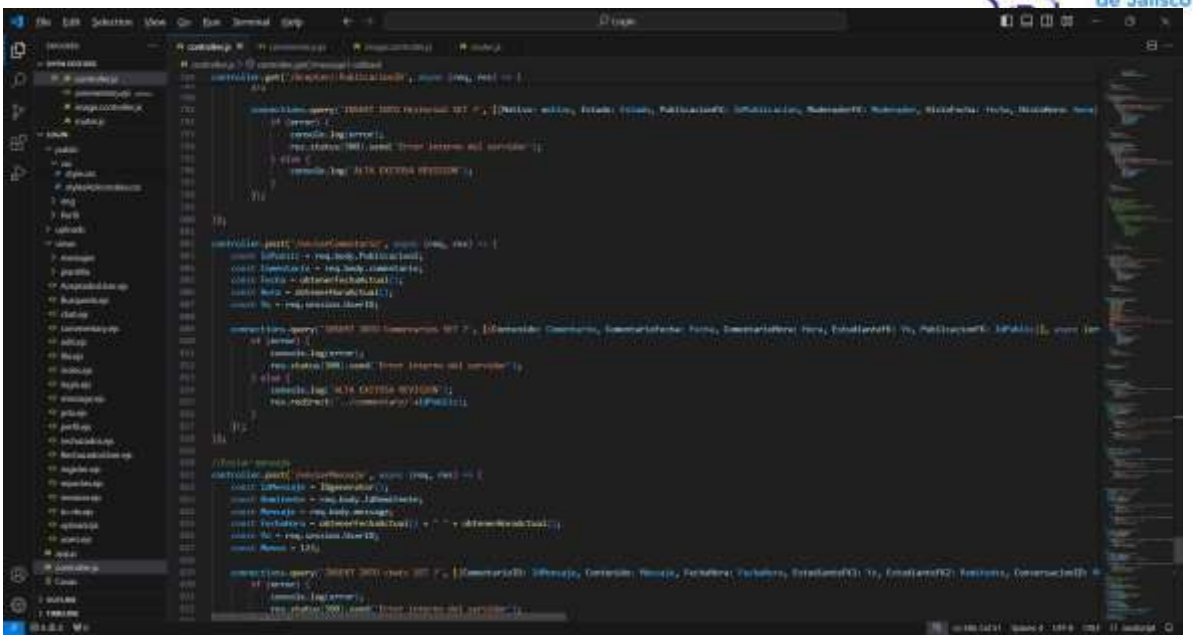


Imagen 72

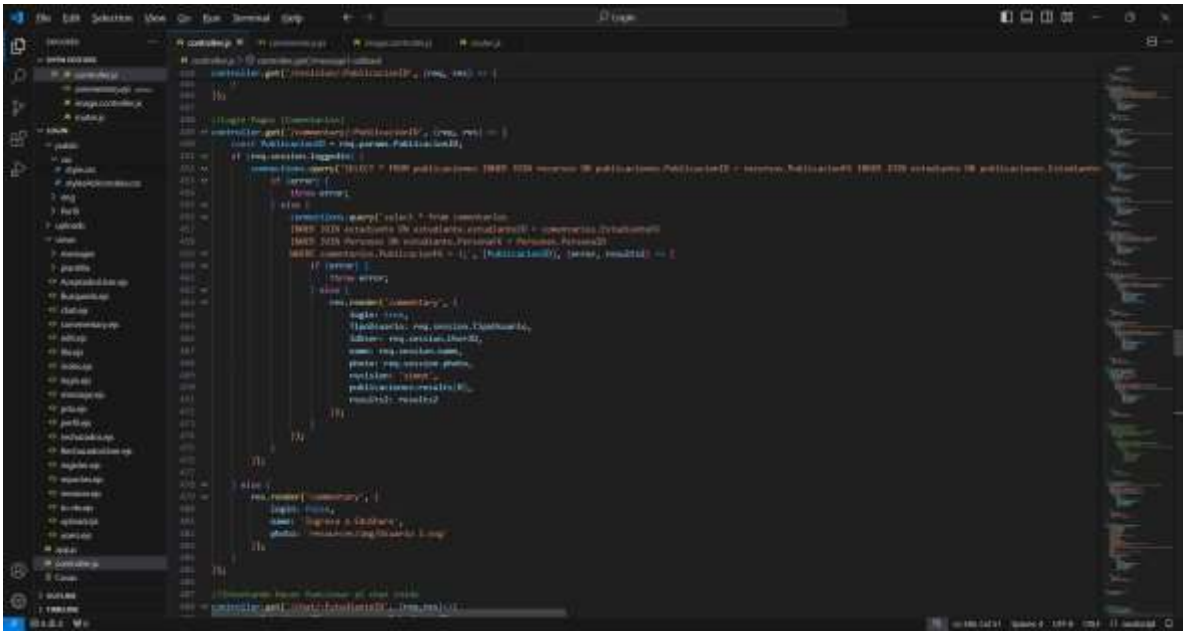


Imagen 73

Reporte

Codigo: En esta parte está el código del modal que utilice, y en la otra parte esta el insert que utilice para meter a la base de datos el reporte y el contenido del reporte

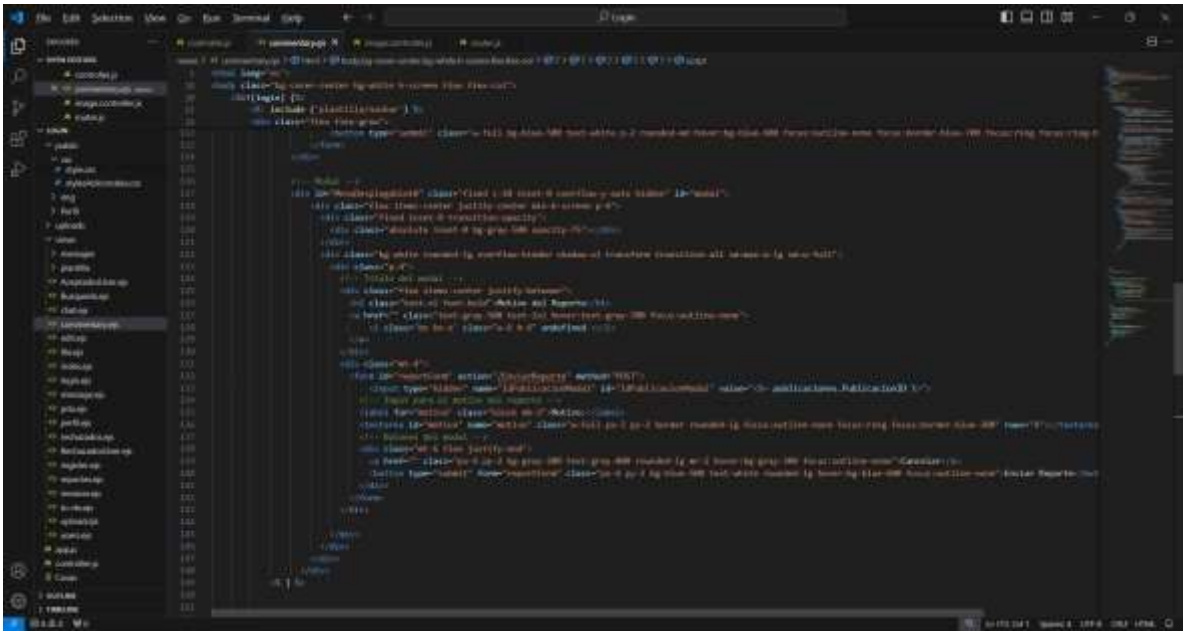


Imagen 74

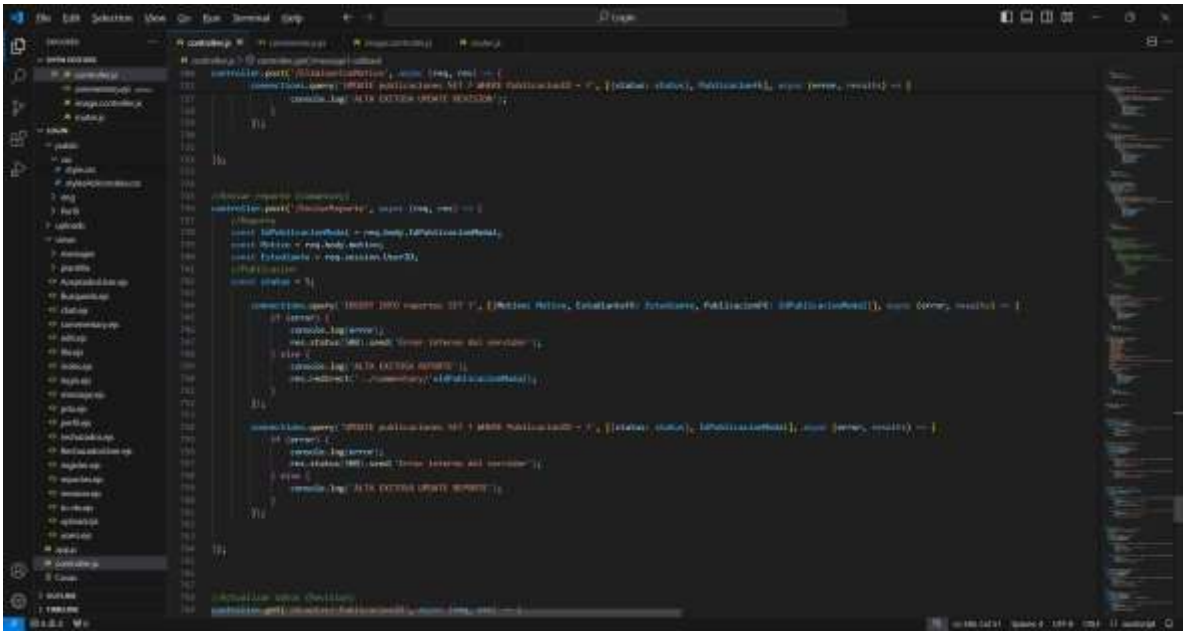


Imagen 75

Solicitudes Pendientes

Codigo: En el código podremos ver toda el front End el cual esta todo el diseño de cada una de estas tarjetas que se muestran en la pantalla, y en la parte del controlador esta las búsquedas en la base de datos para extraer la información

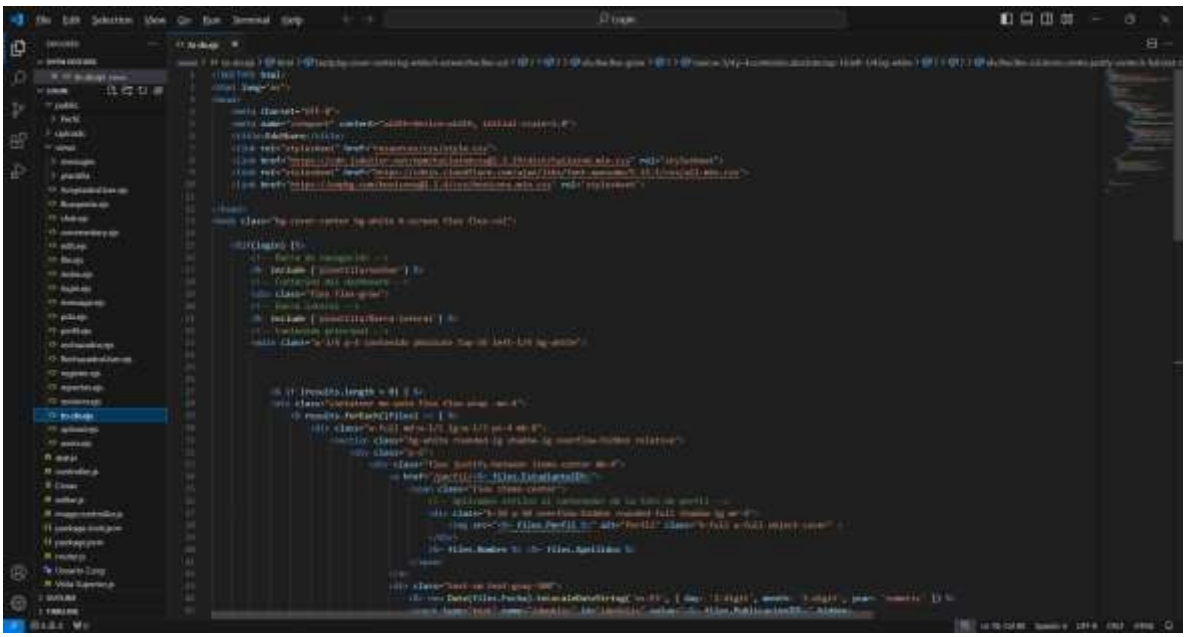


Imagen 76

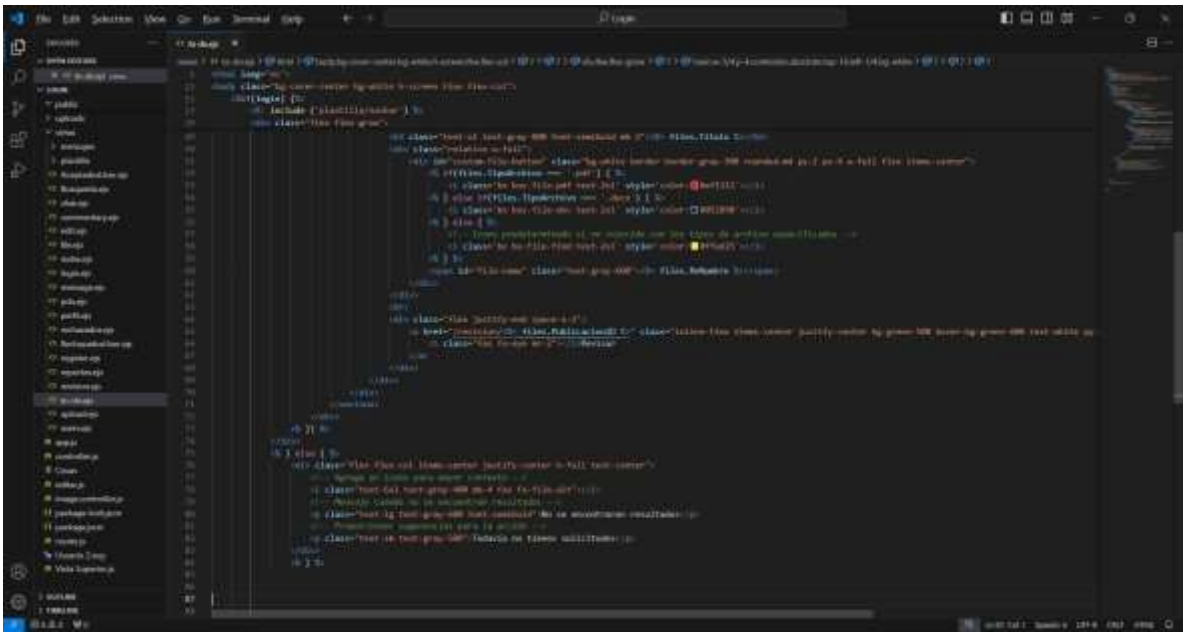


Imagen 77

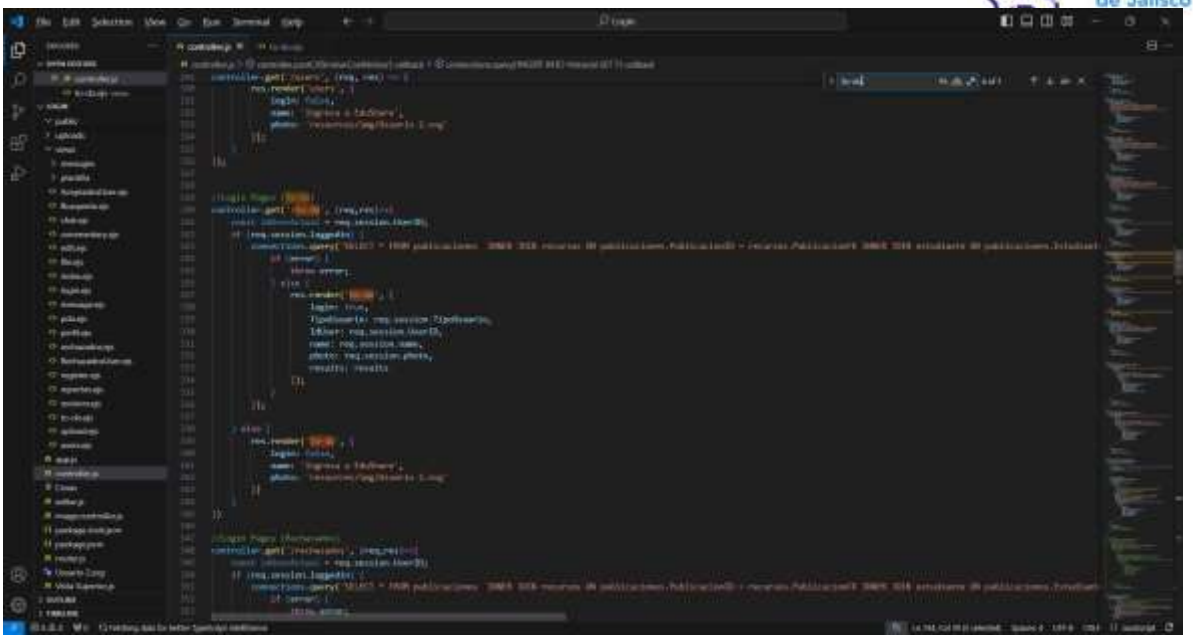


Imagen 78

Revisión

Codigo: En esta parte esta el diseño, la búsqueda para la información coincida una con otra, y la acción del botón aceptar para que se muestre en la pantalla de inicio

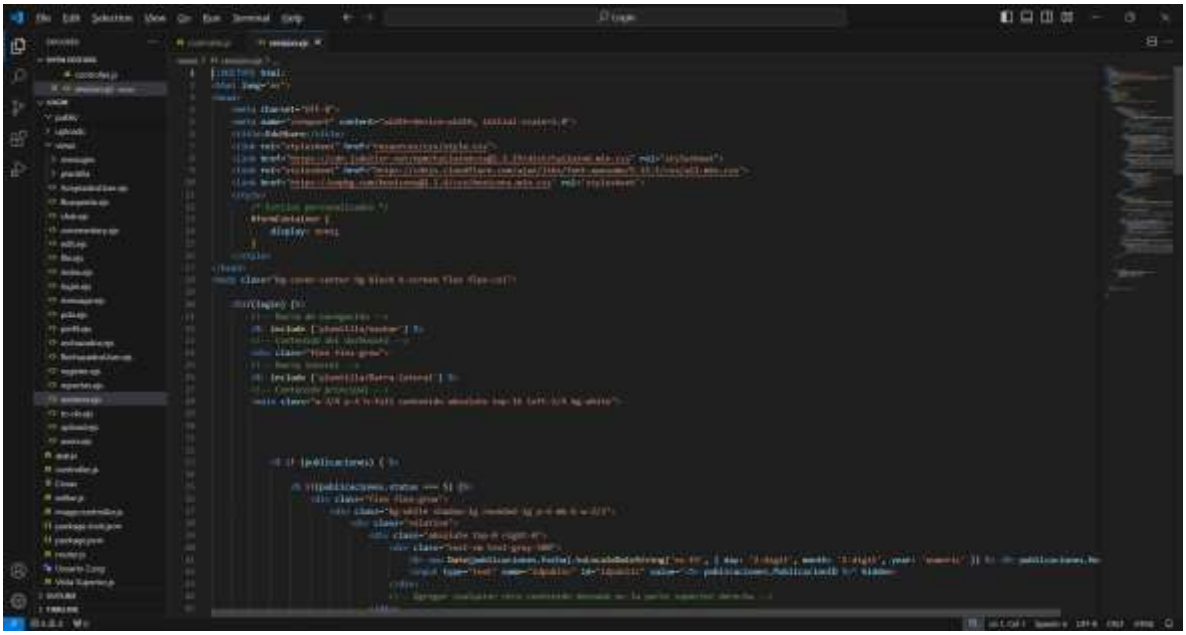


Imagen 79

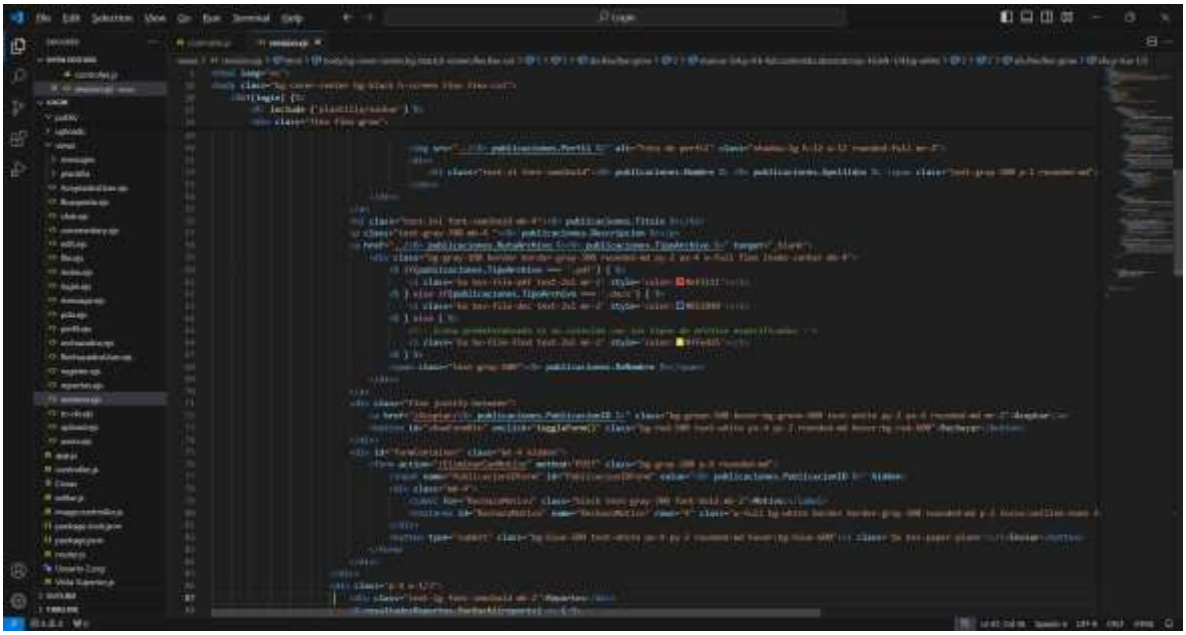


Imagen 80

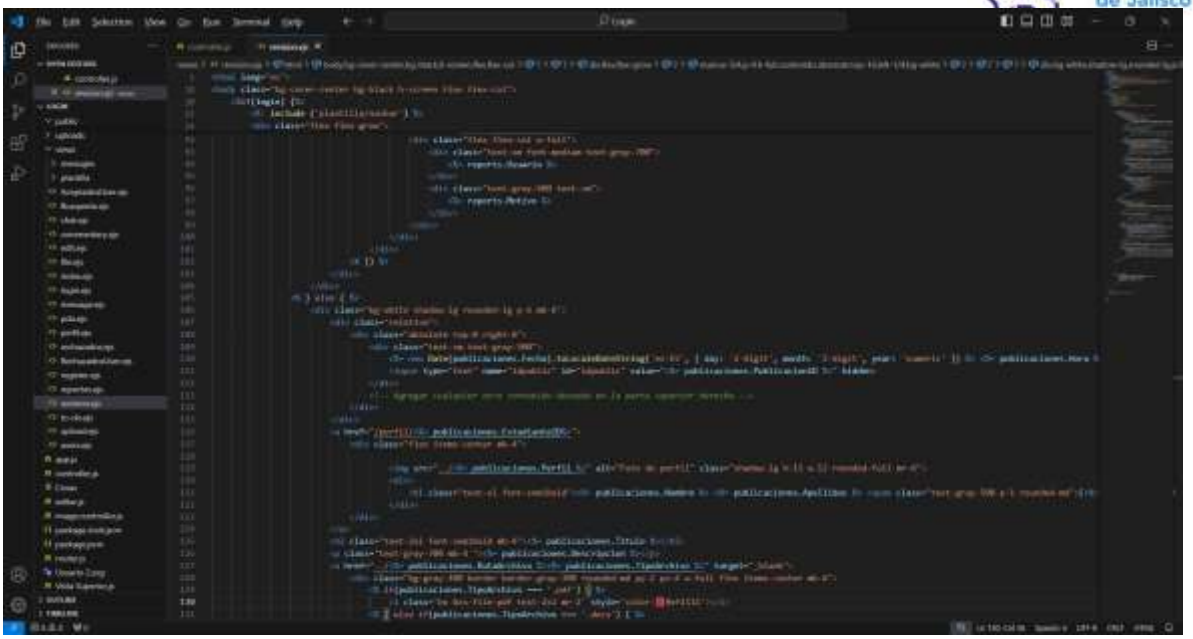


Imagen 81

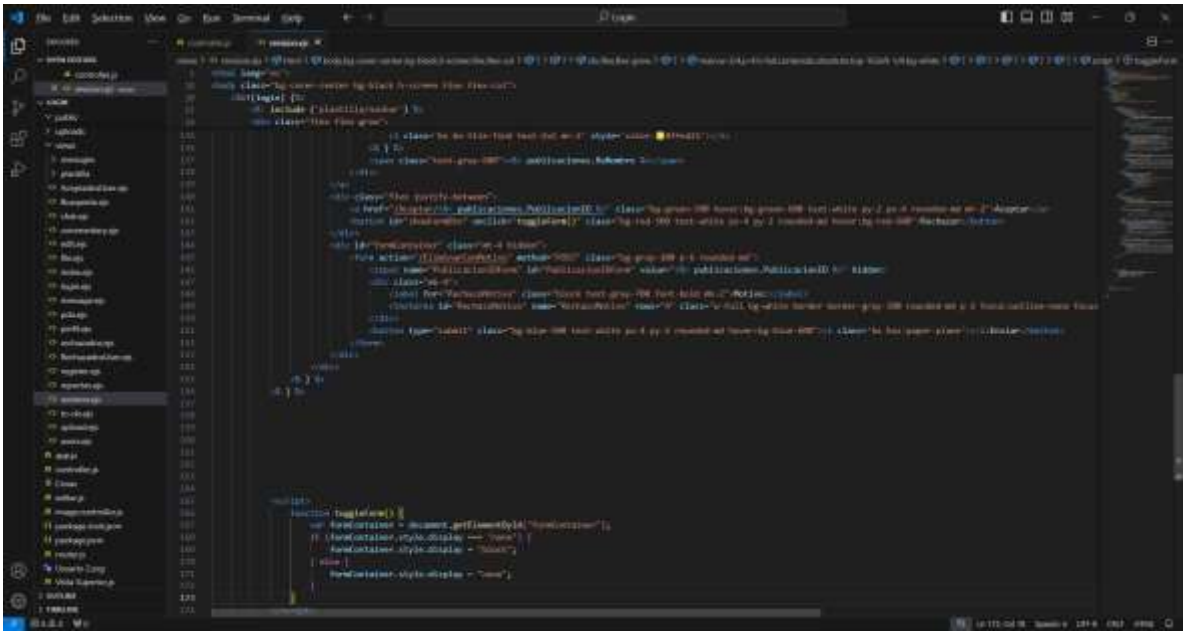


Imagen 82

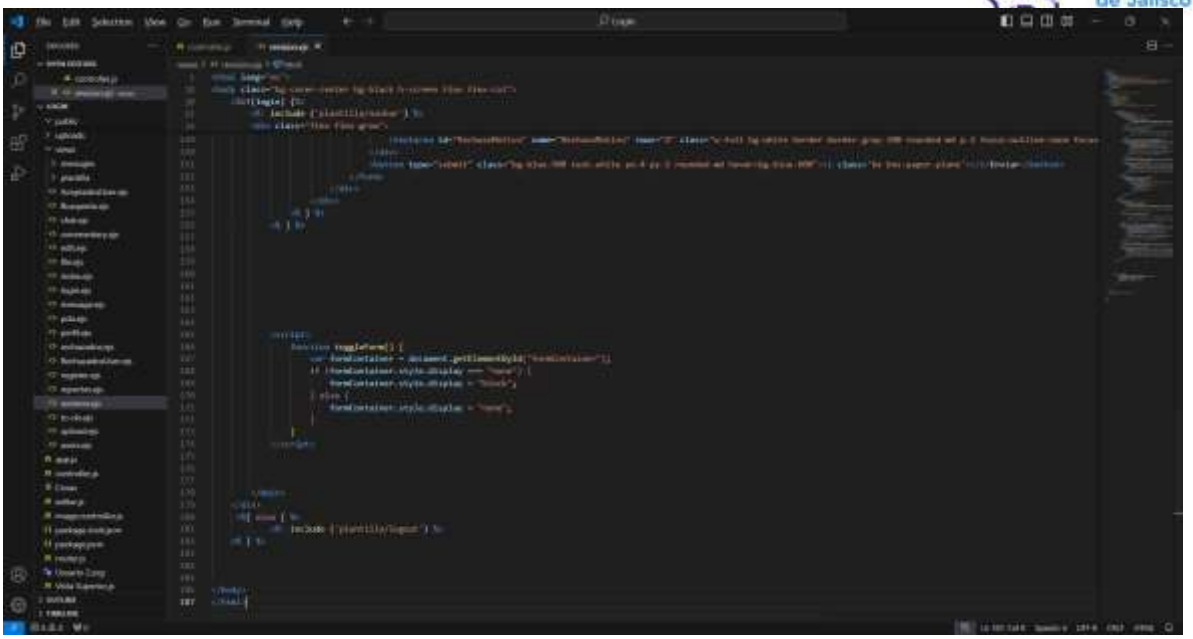


Imagen 83

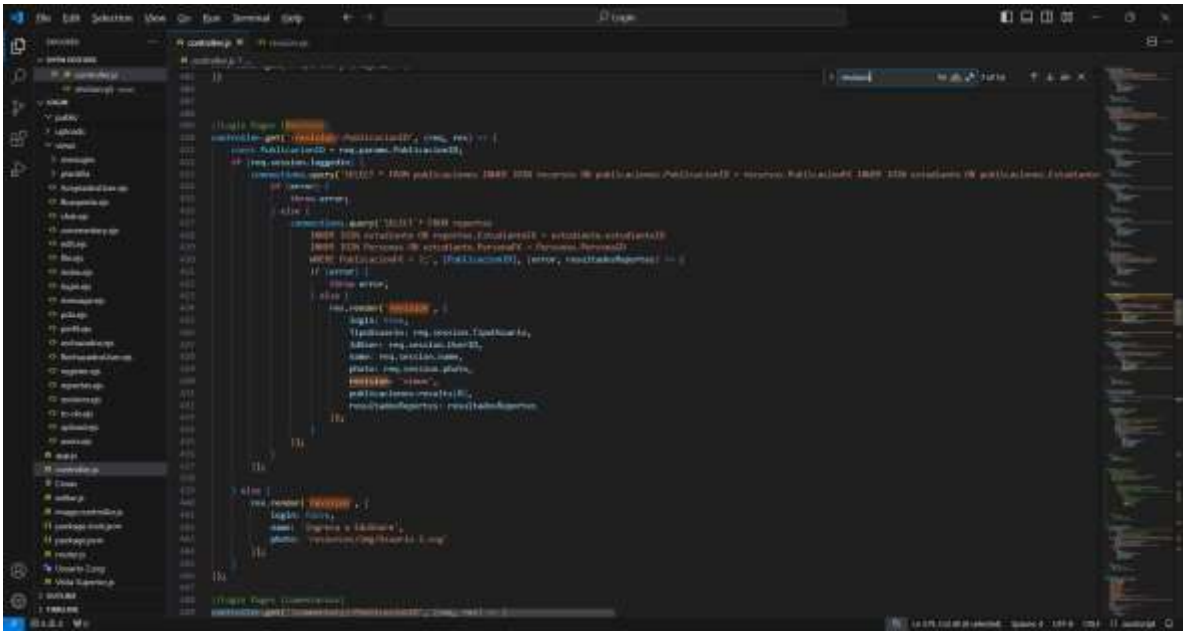


Imagen 84

Rechazo (Formulario)

Codigo: es el mismo que la ventana anterior, sin embargo para este formulario es básicamente un insert extra

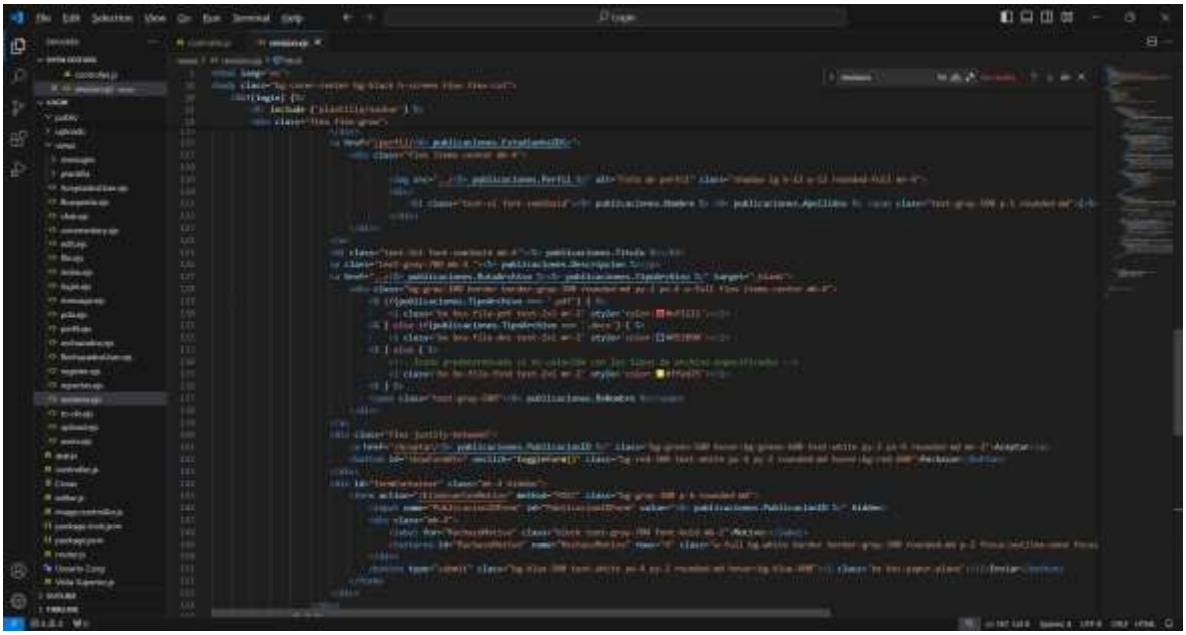


Imagen 85

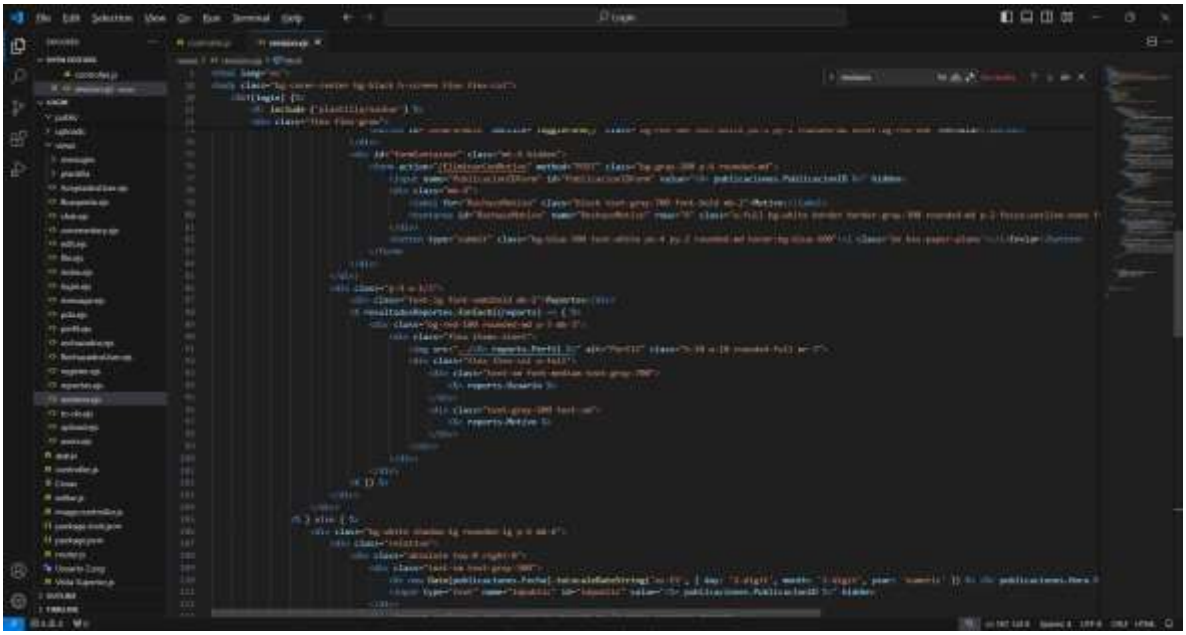


Imagen 86

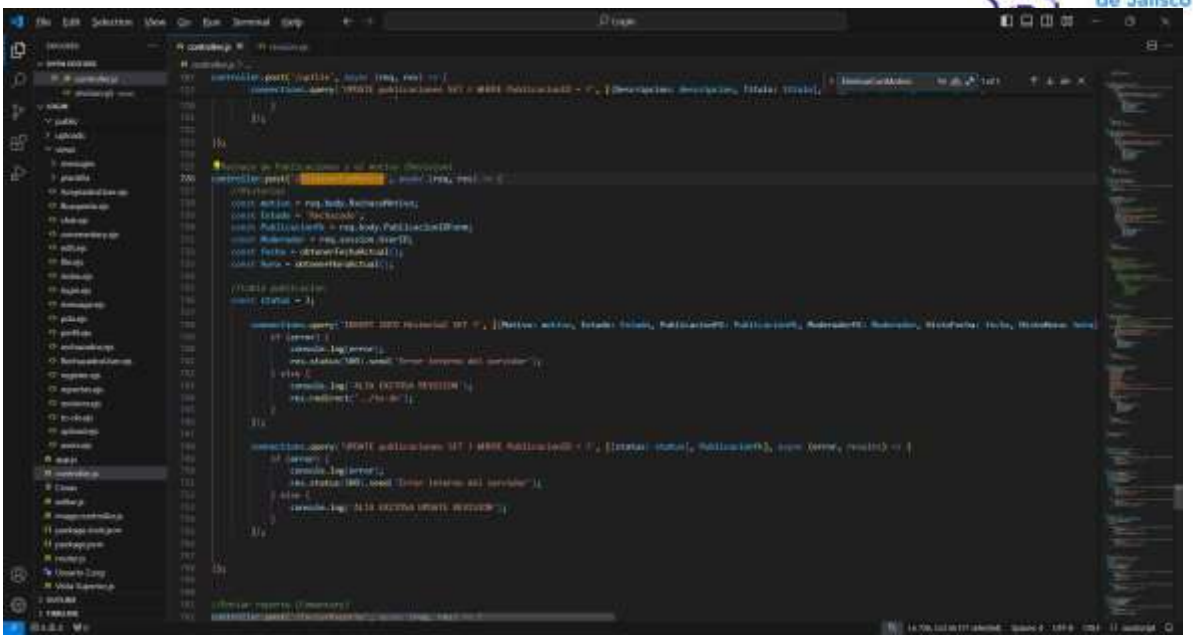


Imagen 87

Rechazados

Codigo: En esta parte esta todo el diseño y la búsqueda en la base de datos para ver todos los rechazados,

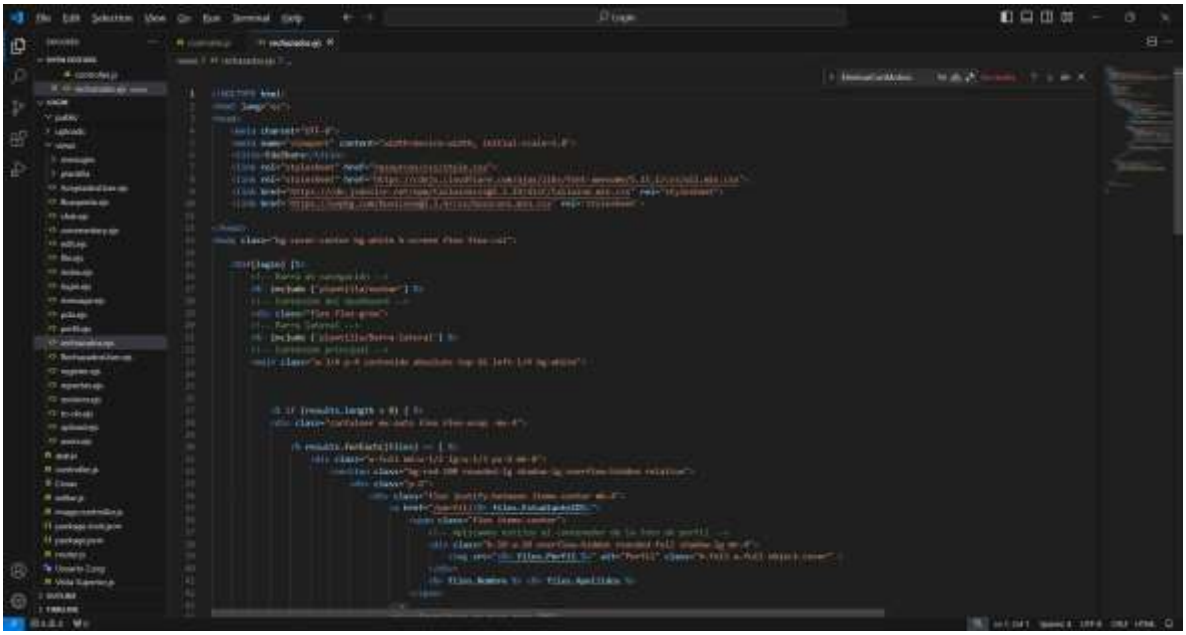


Imagen 88

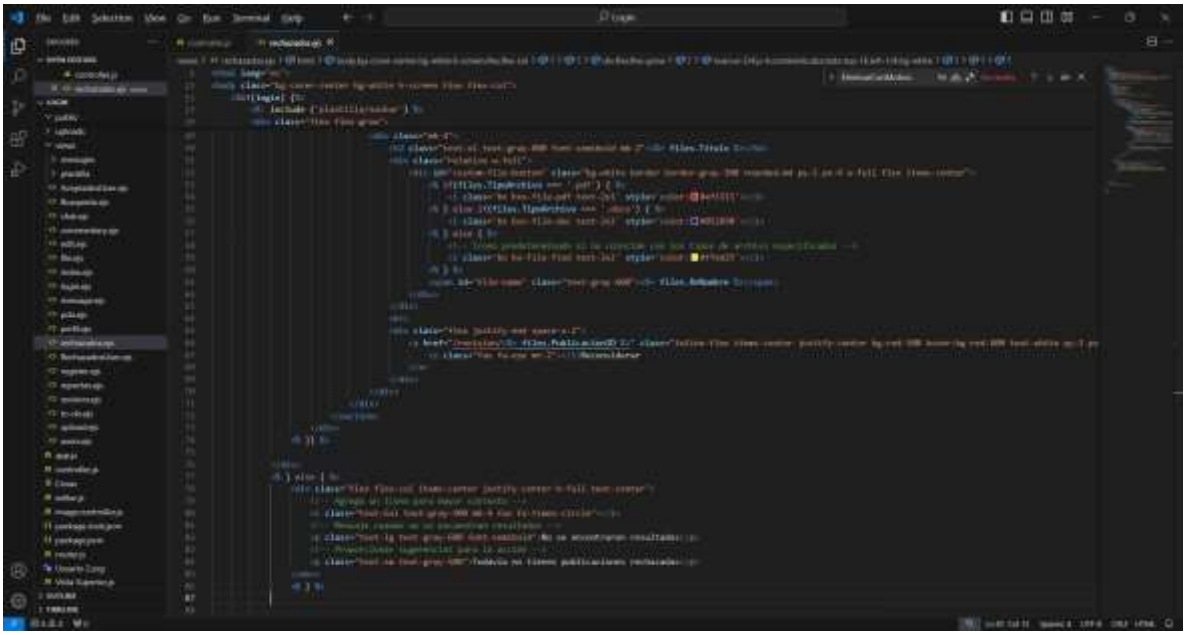


Imagen 89

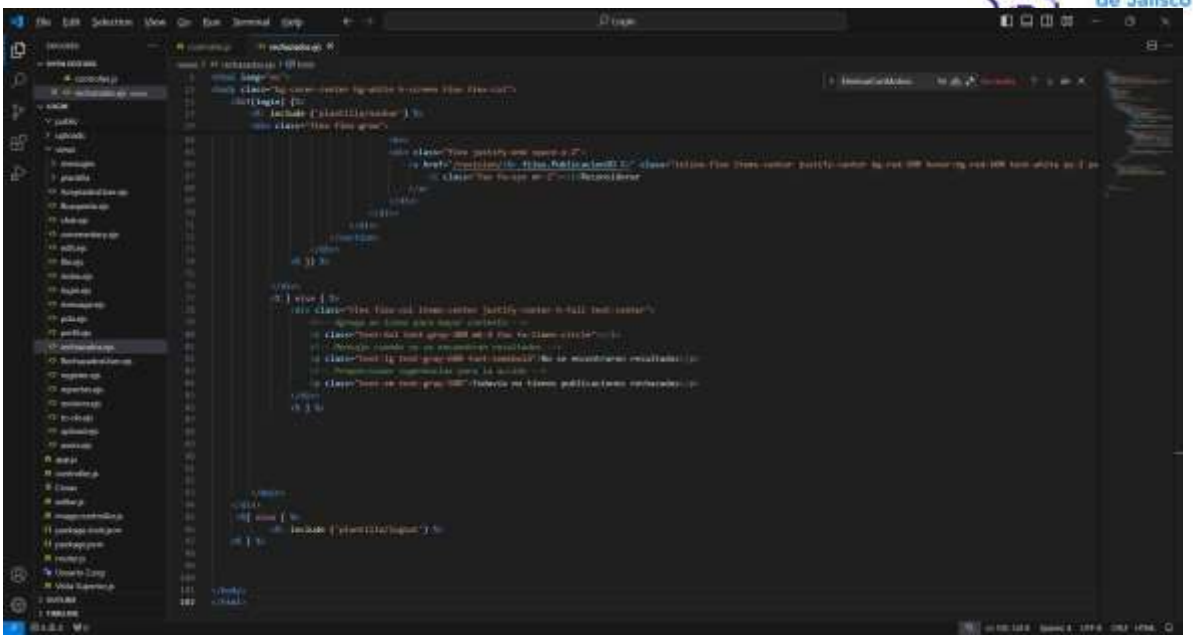


Imagen 90

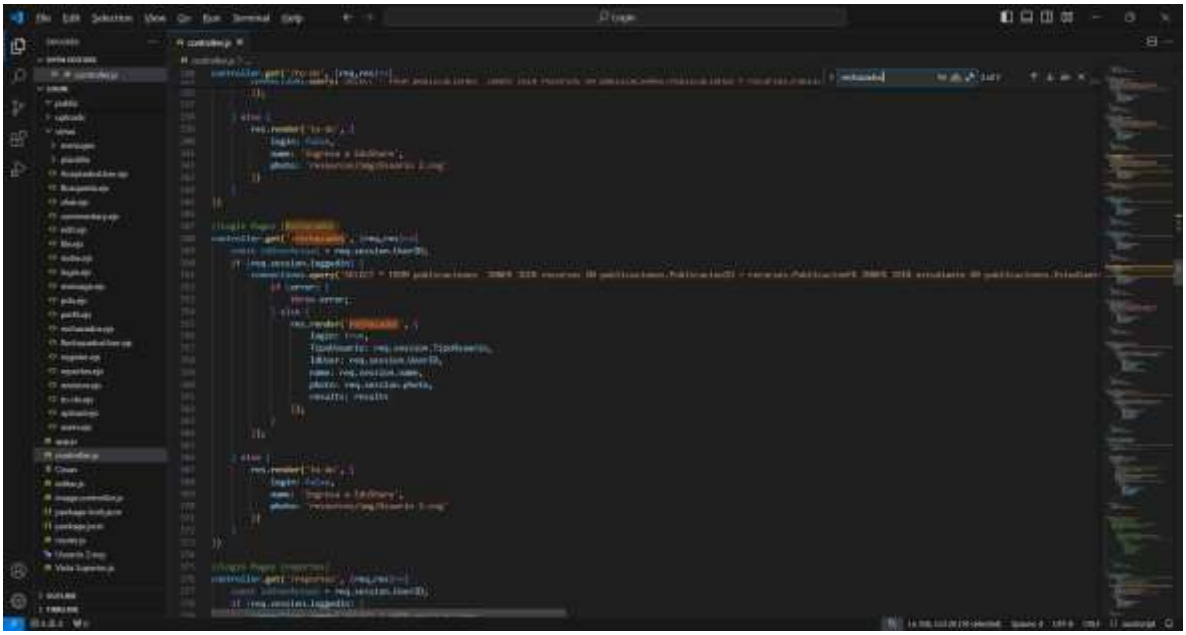


Imagen 91

Reportes

Código: Esta el diseño de la ventana y la búsqueda de los reportes existentes, aquí solo vemos lo que fueron reportados antes de revisar si vale la pena considerar el reporte, si lo aceptamos como ni no pasara nada o lo eliminamos y rechazamos

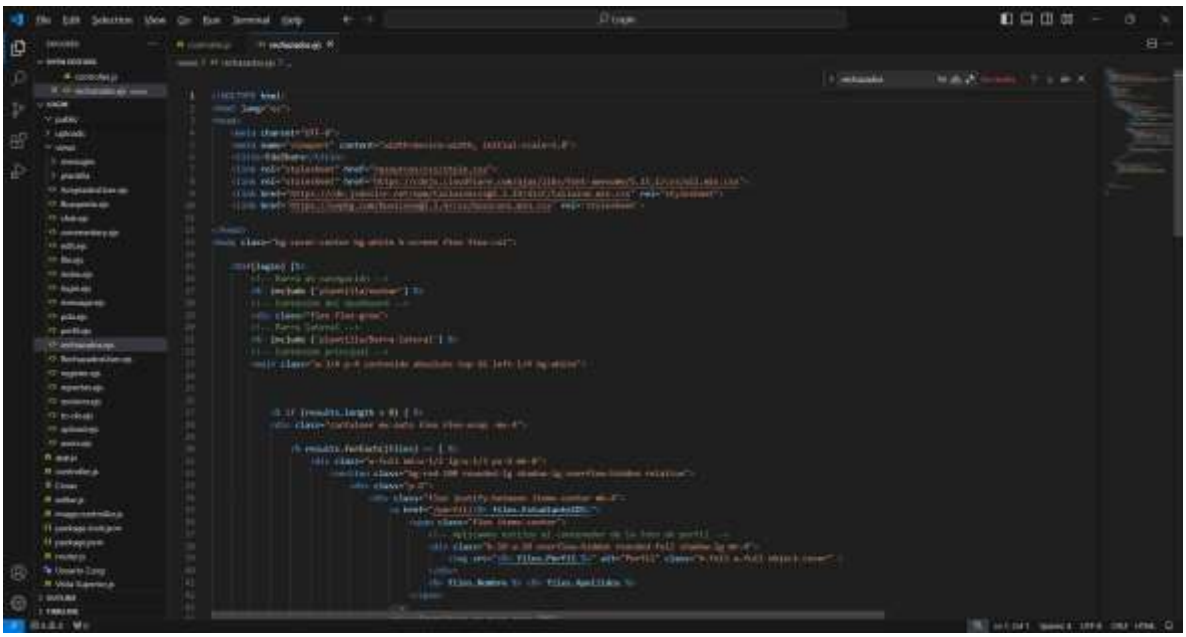


Imagen 92

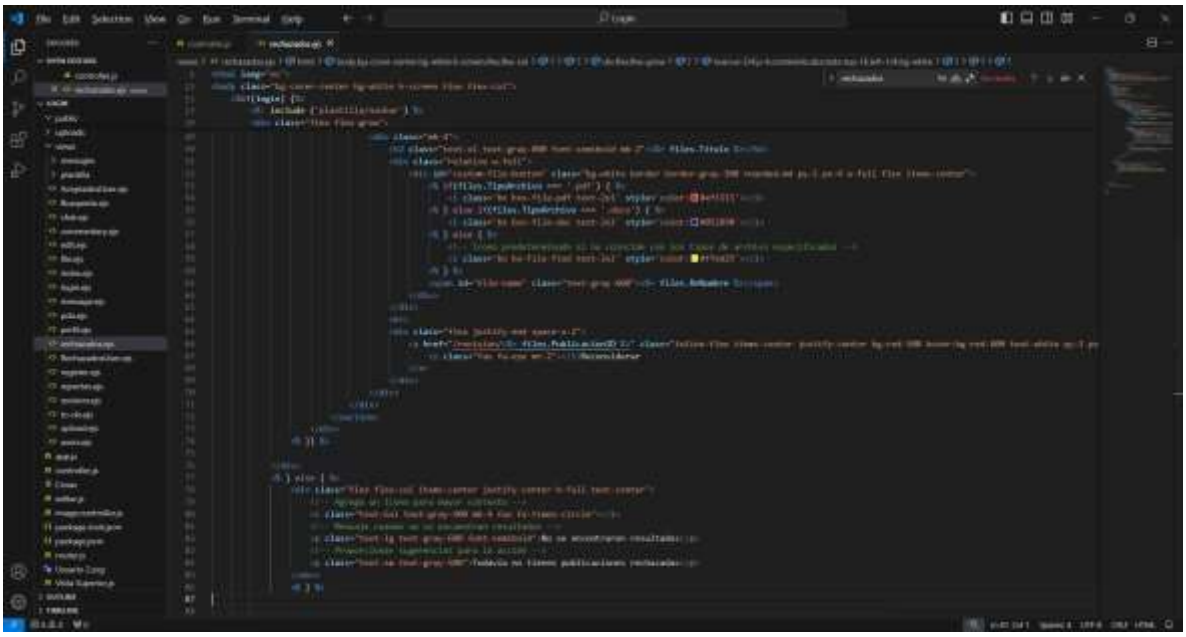


Imagen 93

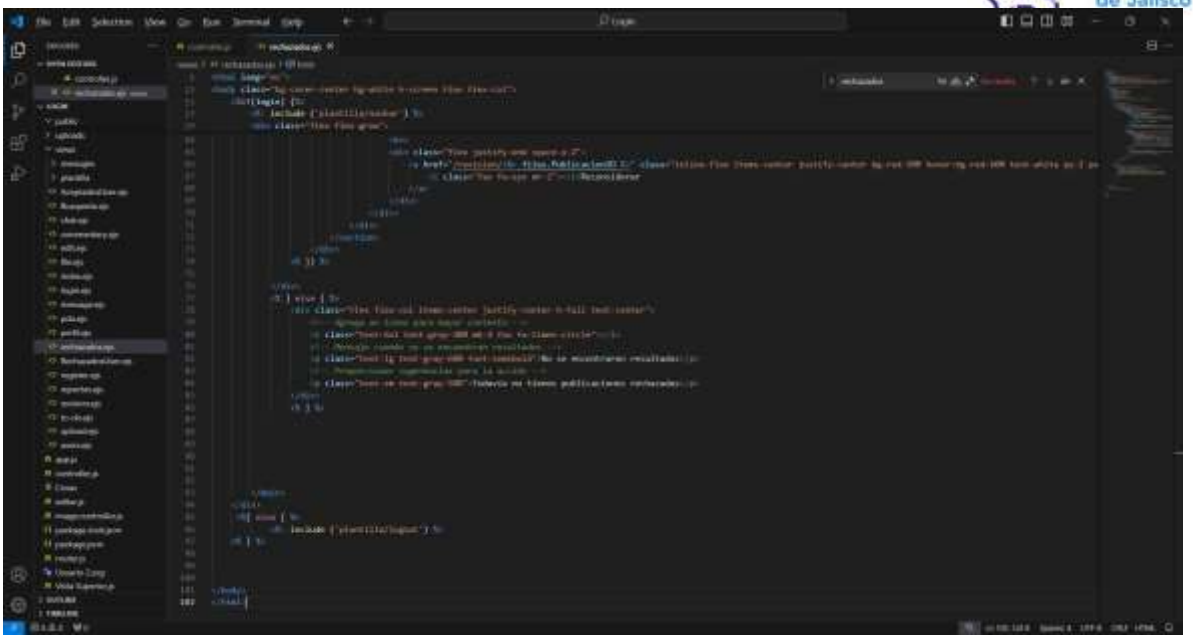


Imagen 94

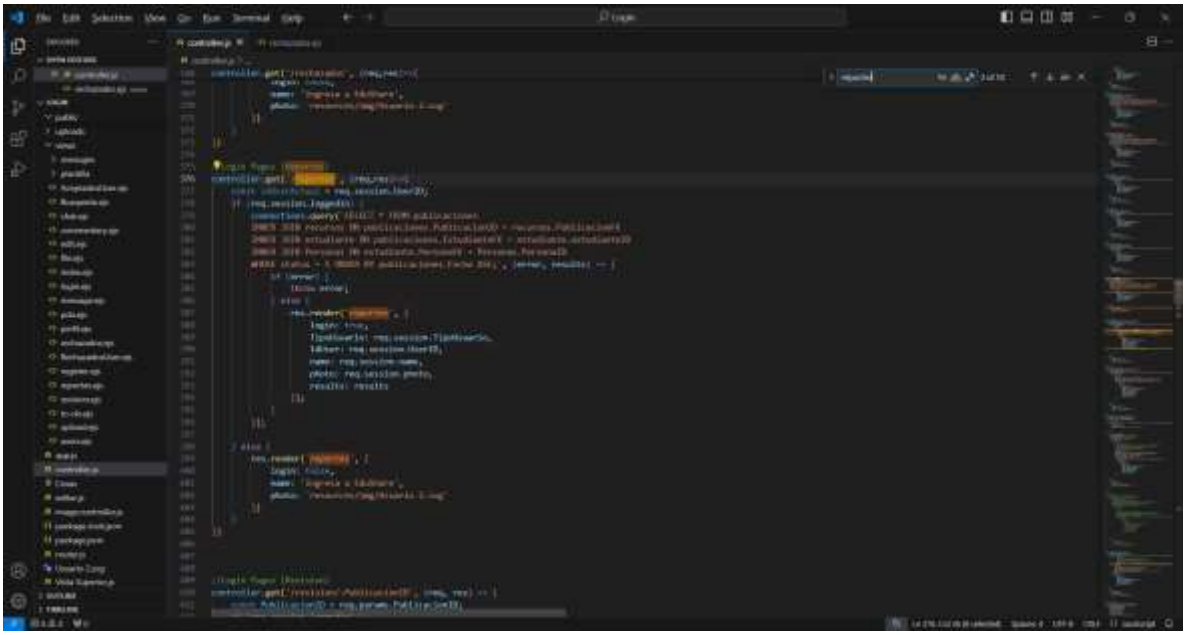


Imagen 95

Revisión de reportes

Codigo: En esta parte esta el diseño, la búsqueda para la información coincida una con otra, y la acción del botón aceptar para que se muestre en la pantalla de inicio

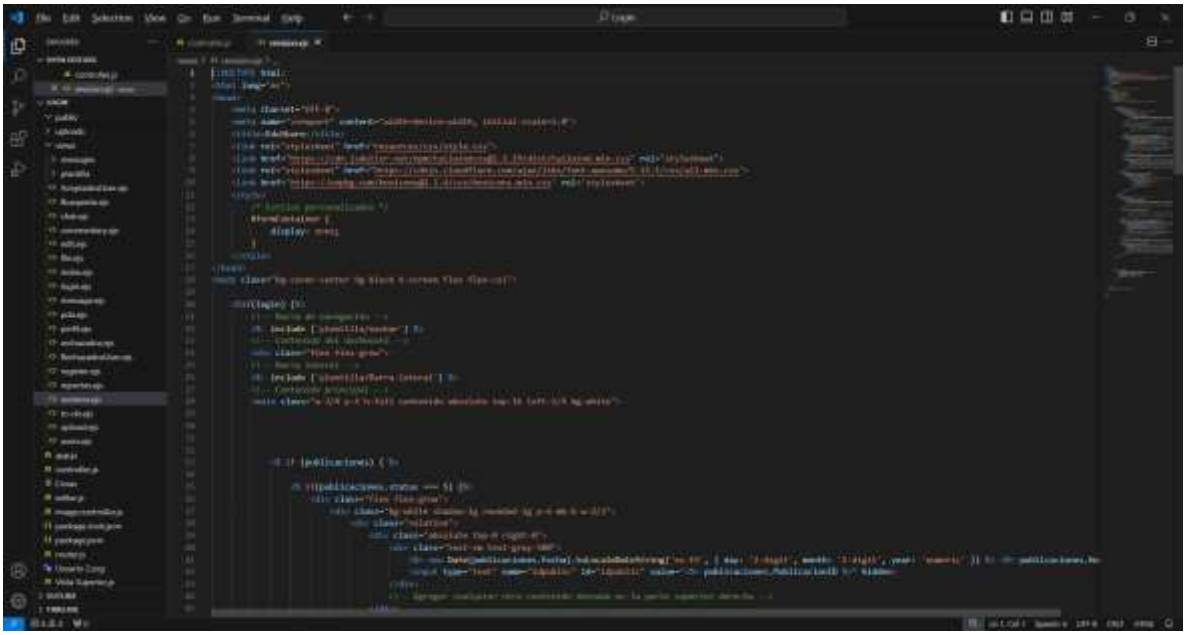


Imagen 96

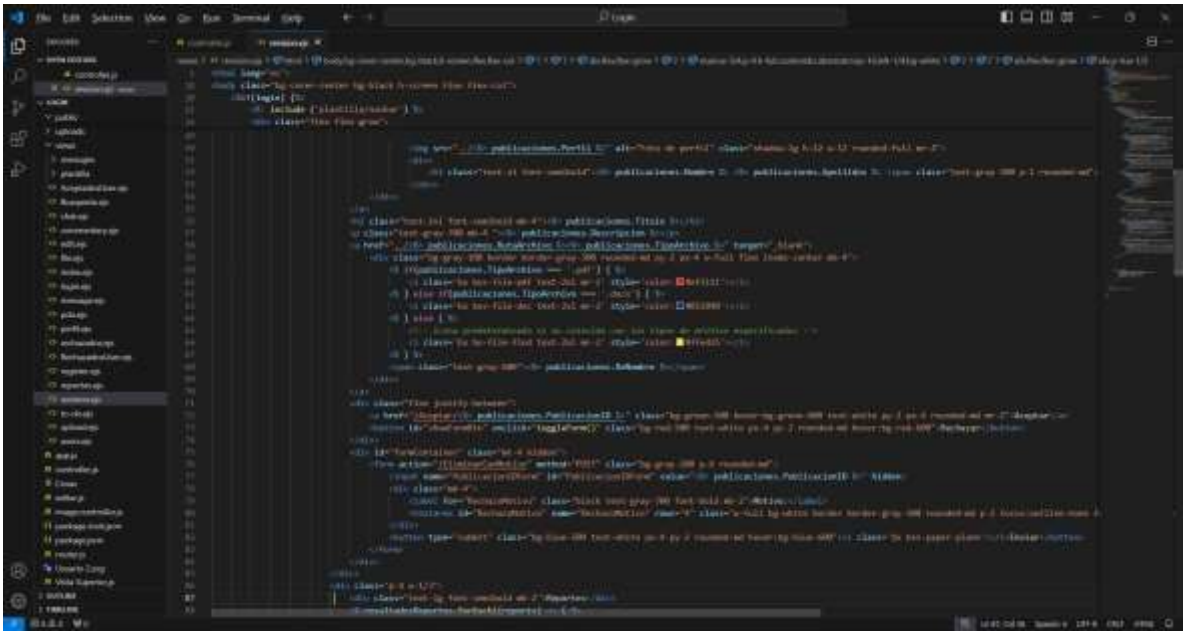


Imagen 97

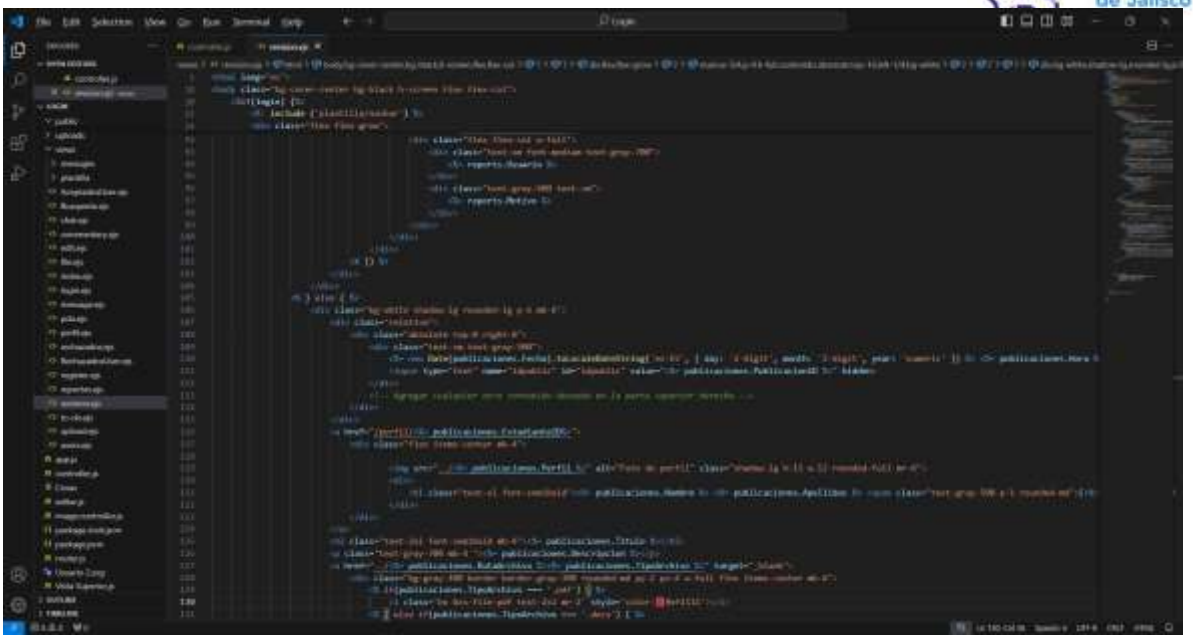


Imagen 98

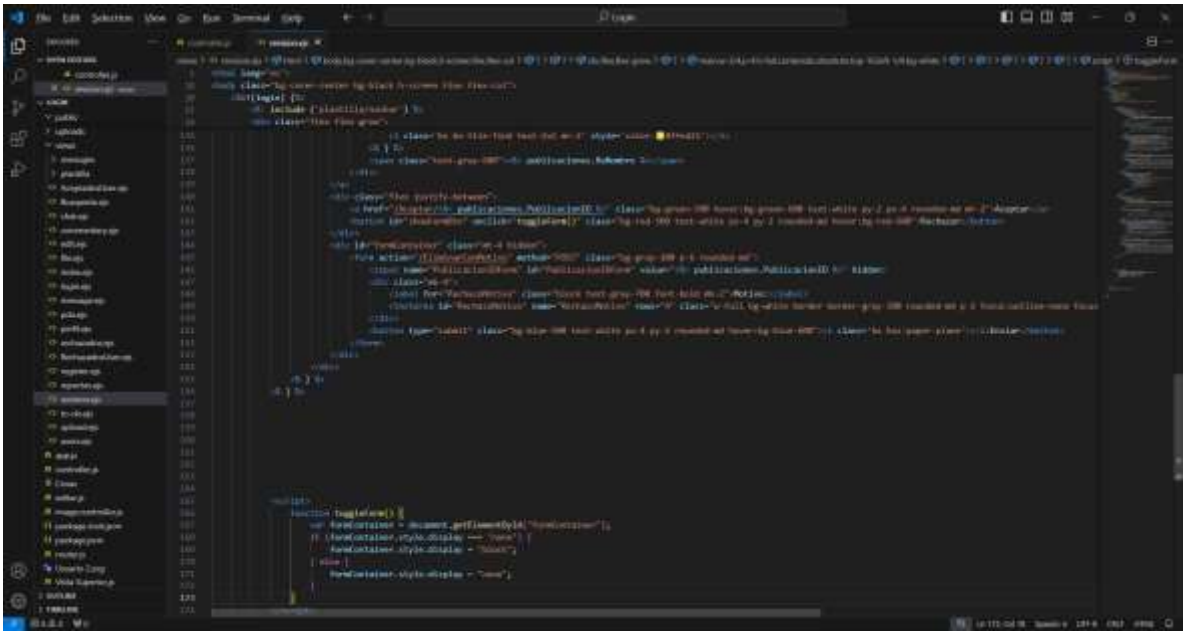


Imagen 99

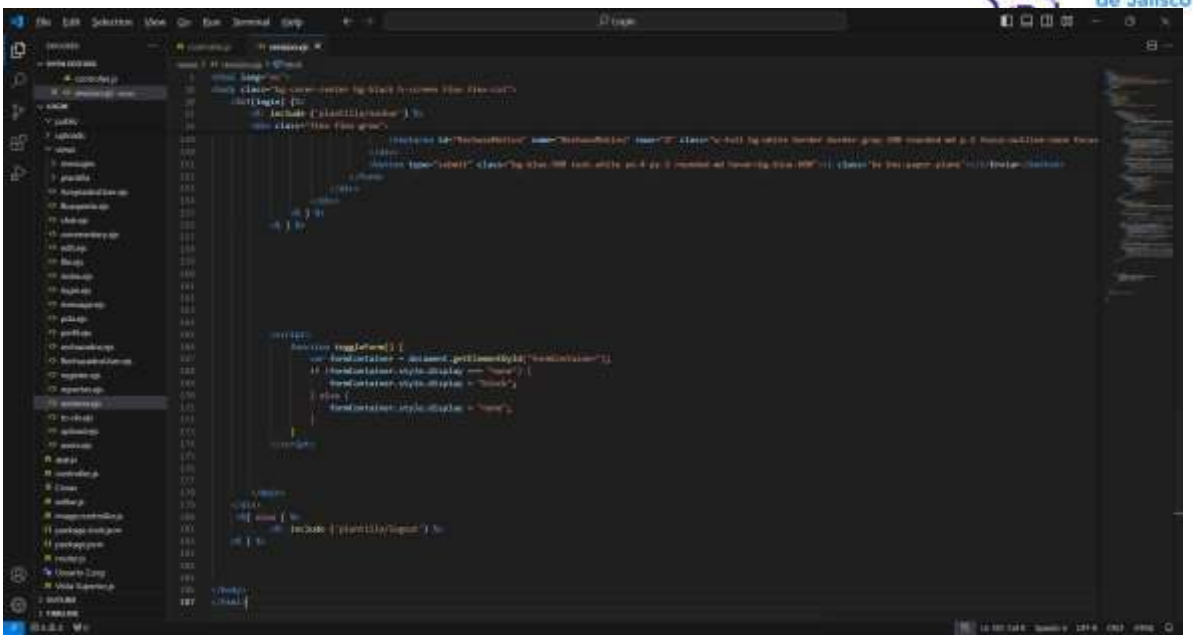


Imagen 100

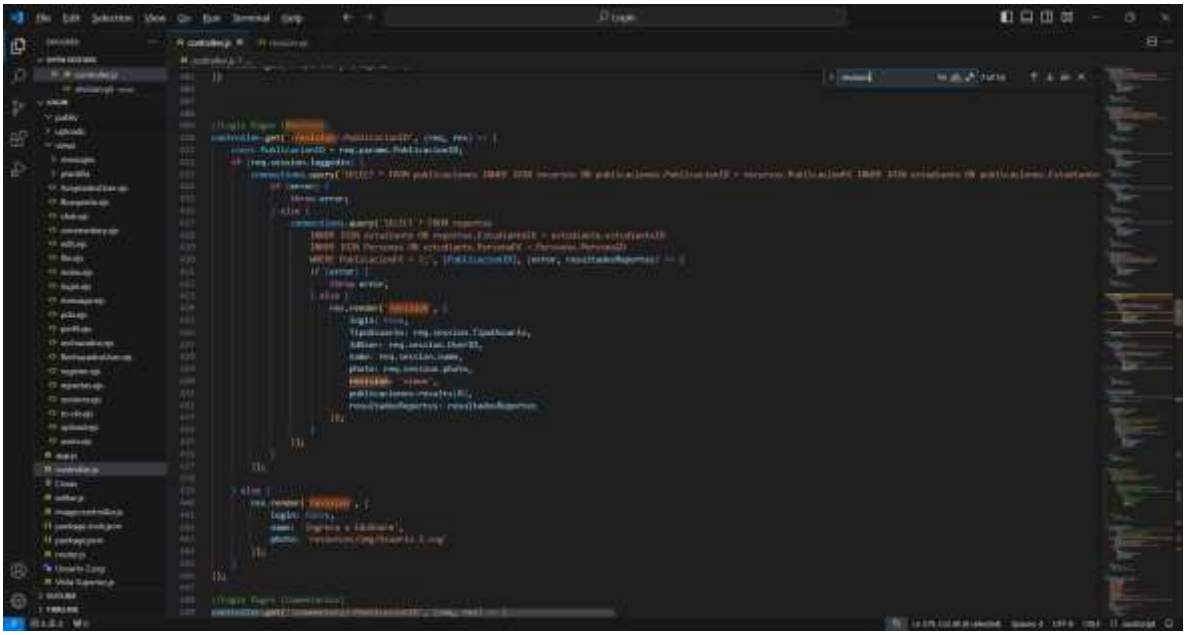


Imagen 101

Rechazo (Formulario)

Codigo: es el mismo que la ventana anterior, sin embargo para este formulario es básicamente un insert extra

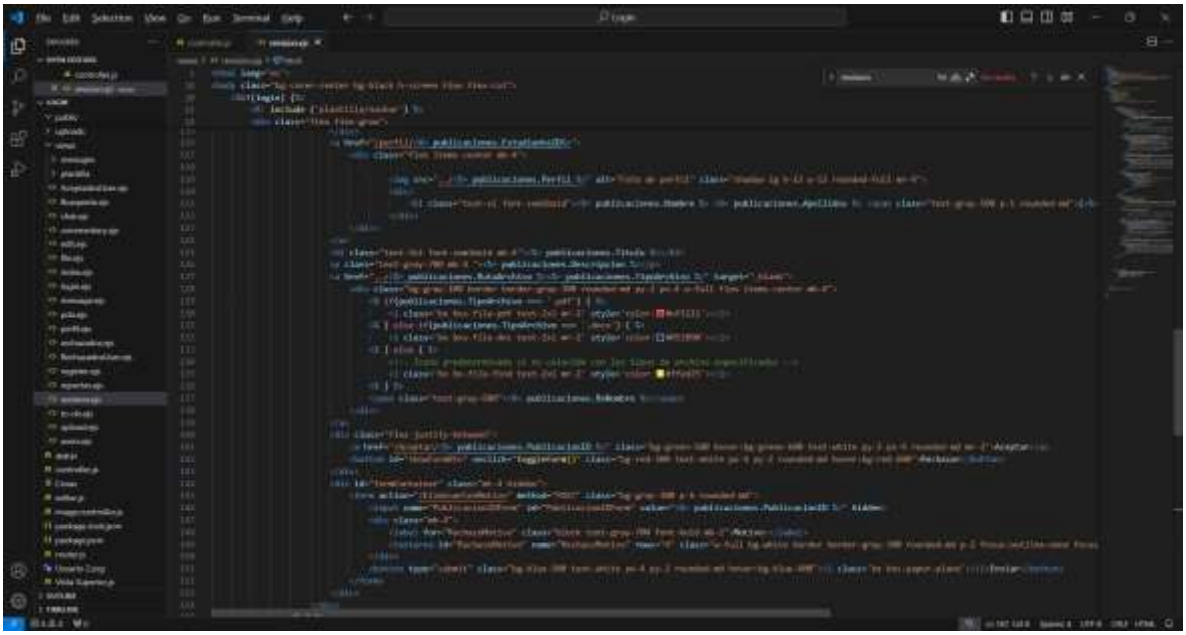


Imagen 102

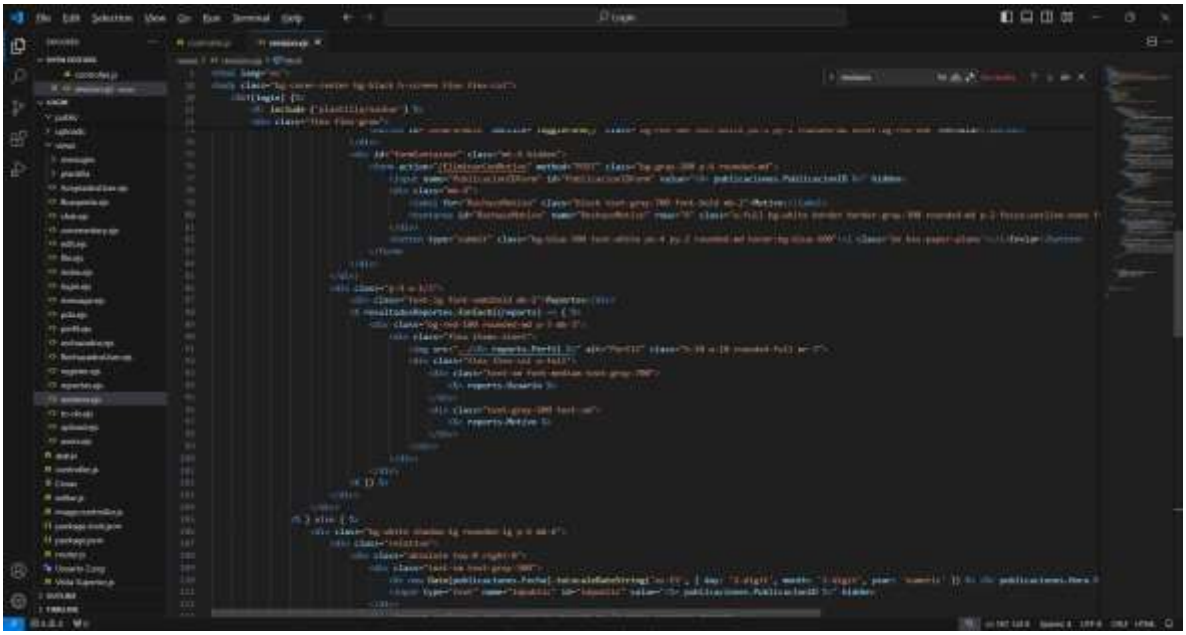


Imagen 103

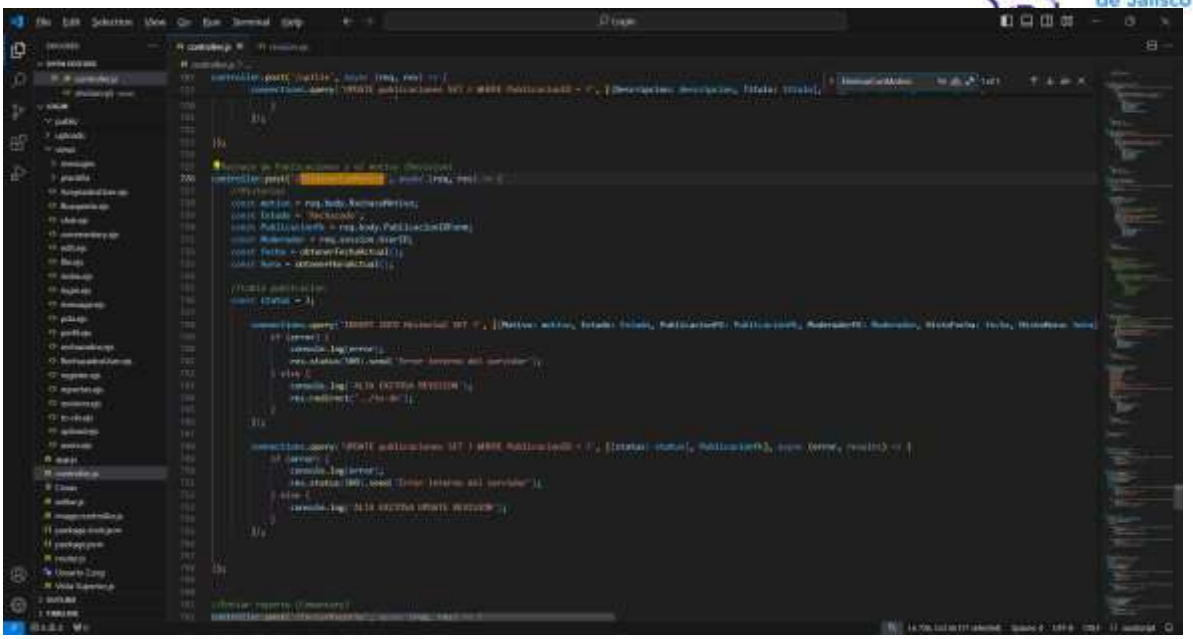


Imagen 104

Control de usuarios

Codigo: En esta parte esta toda la gestión de usuarios, esta el CRUD completo, puedes editar y mejorar lo que quieras

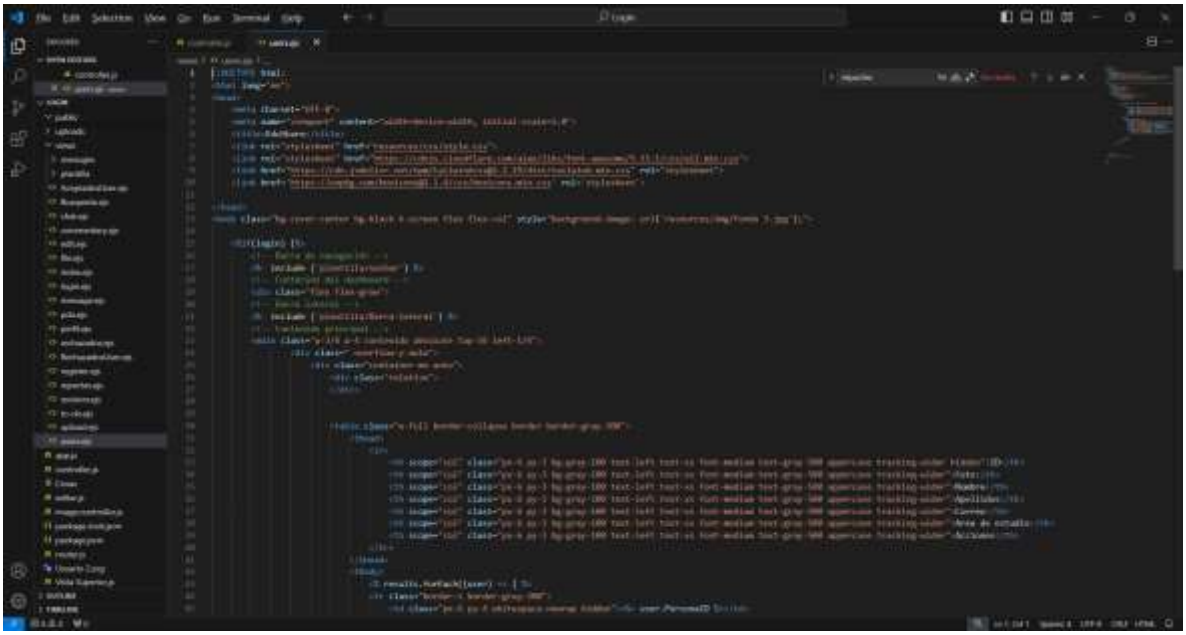


Imagen 105

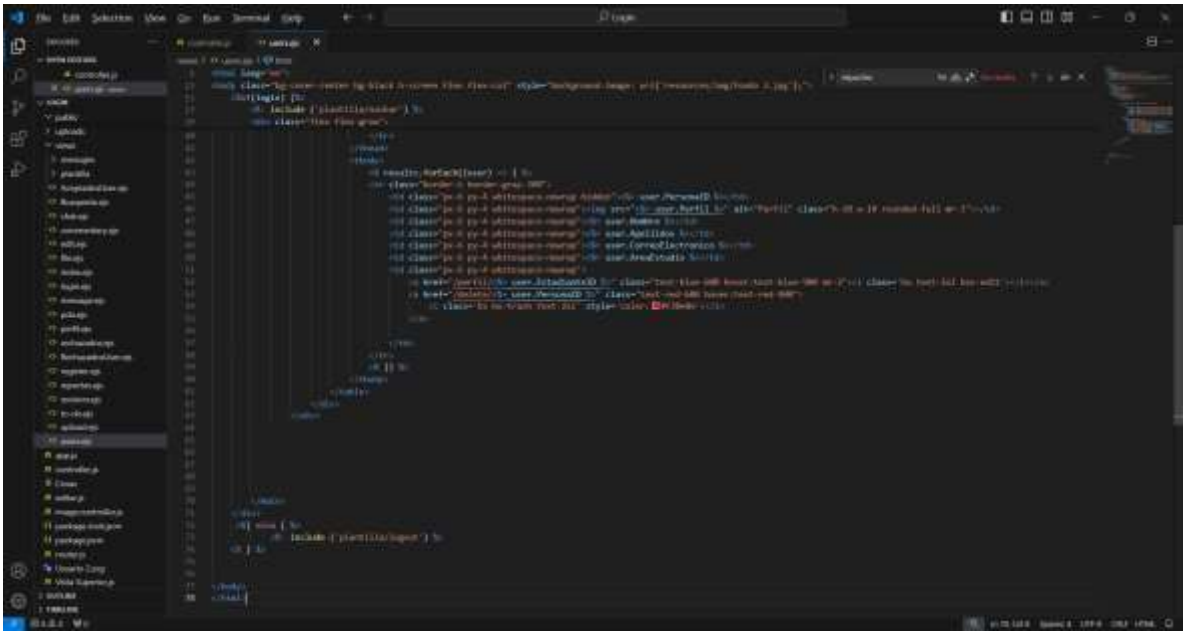


Imagen 106

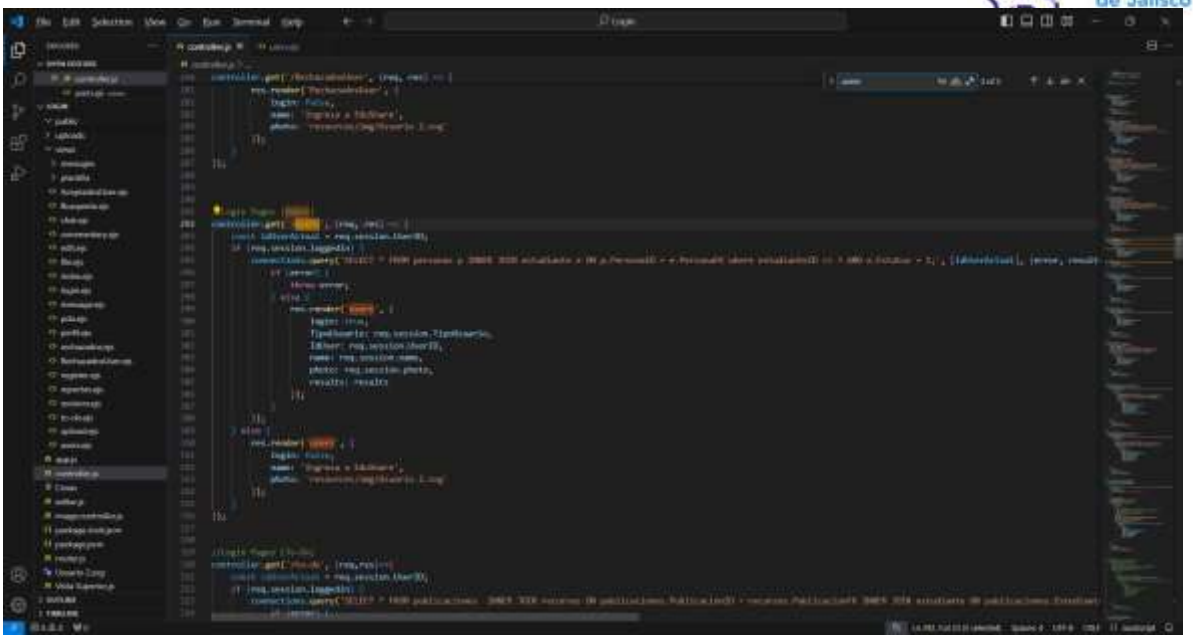
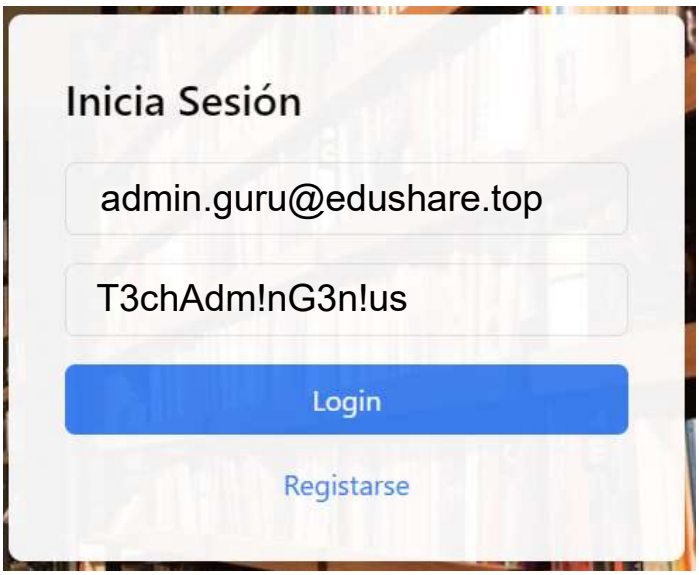


Imagen 107

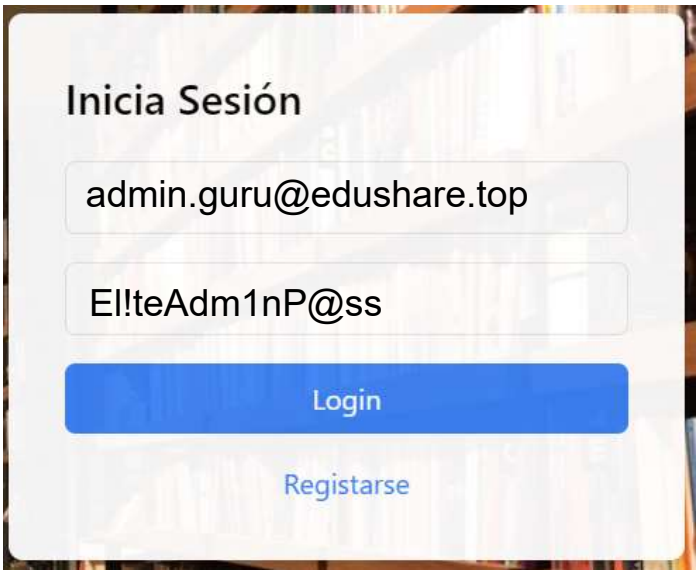
Link -> Usuario y Contraseña

Administrador (Por diseño lo puse en una imagen, pero se puede copiar y pegar el texto):

admin.guru@edushare.top



Usuario



Competencias

Bibliografías (Formato APA)

- **Sebastopol, A. (2021). Programación para principiantes: Aprende a programar con Python desde cero.** https://www.amazon.com.mx/Python-Para-Principiantes-Aprender-programar/dp/B0BRH4KG4F
- **Gutiérrez, J. L. (2020). Fundamentos de programación.** https://www.mheducation.com.mx/fundamentos-de-programacion-9786071514684-latam-group
- **Eckel, B. (2020). Pensamiento computacional: Conceptos y ejercicios en Python.** https://www.amazon.com.mx/Introducci%C3%B3n-programaci%C3%B3n-Python-Algoritmos-principiantes-ebook/dp/B072YX6K8R**Pressman, R. S., & Software Quality Institute. (2020). Ingeniería de software: Un enfoque ágil (10a ed.). McGraw-Hill Interamericana.** https://www.amazon.com/-/es/Roger-S-Pressman/dp/8476152221
- **Sommerville, I. (2021). Ingeniería de software (10a ed.). Pearson Educación.** https://www.amazon.com/-/es/Ian-Sommerville/dp/6073206038
- **Larman, C. R. (2021). Ingeniería de software ágil: Principios, patrones y prácticas.** https://www.amazon.com.mx/Ingenieria-Software-Sommerville/dp/6073206038**Larman, C. R., & Basili, V. R. (2021). Metodologías ágiles de desarrollo de software.** https://www.amazon.com.mx/Metodolog%C3%ADa-%C3%A1gil-principiantes-principios-Publishing/dp/1654152692
- **Cohn, S. (2020). Scrum: La guía definitiva para el desarrollo de software ágil.** https://www.amazon.com.mx/Scrum-Metodologia-Gestionar-Desarrollo-Productos/dp/1724833014
- **Highsmith, A. (2020). Desarrollo de software ágil utilizando Kanban: Un enfoque visual para el flujo de trabajo y la gestión de la calidad.** https://www.amazon.com/Kanban-definitiva-metodolog%C3%ADa-desarrollo-software/dp/164748281X**PMI. (2019). A Guide to the Project Management Body of Knowledge (PMBOK Guide) (6th ed.). Project Management Institute.** https://www.pmi.org/pmbok-guide-standards/foundational/pmbok
- **Wysocki, R. K., & Le Blanc, R. (2020). Effective project management: From planning to execution.** https://www.amazon.com.mx/b?ie=UTF8&node=9575892011
- **Kerzner, H. B. (2020). Project management: A case study approach.** https://www.amazon.com.mx/Project-Management-Approach-Scheduling-Controlling/dp/1119805376**Myers, G. J., & Forta, D. (2020). The art of software testing (4th ed.). Addison-Wesley.** https://www.amazon.com/Art-Software-Testing-Glenford-Myers/dp/B002W6HZ1W
- **Pezze, M., & Young, M. (2020). Software testing: A craft tester's perspective (4th ed.). Pearson Education.** https://www.amazon.com/Software-Testing-Craftsmans-Approach-Fourth/dp/1466560681

- **Galin, D. (2020). Bad code: How to write it and how to avoid it.** https://www.amazon.com/How-Not-Write-Bad-Problems/dp/1594488487**Humble, J., & Farley, J. (2020). Continuous delivery: Reliable software releases at scale.** https://www.amazon.com/-/es/Jez-Humble/dp/0321601912
- **Ulloa, M. (2020). DevOps: La integración del desarrollo y las operaciones en la era digital.** https://www.amazon.com/-/es/Ethan-Thorpe/dp/1696089077
- **Feiler, P., & Humphrey, S. (2020). Automated software testing: A practical guide.** https://www.amazon.com/software-testing/s?k=software+testing**Frink, T. (2021). Tecnología para principiantes: Una guía básica para comprender el mundo digital.** https://www.amazon.com.mx/b?ie=UTF8&node=21393546011
- **O'Brien, J. A., & Marakas, G. M. (2020). Management information systems (10th ed.). McGraw-Hill Education.** https://www.amazon.com/Management-Information-Systems/b?ie=UTF8&node=10806574011
- **Buyya, R., & Marimuthu, P. (2020). Cloud computing: Principles and paradigms.**

Anexos

Carta Constitutiva

Contrato de EduShare

Autor	Versión	Fecha Inicio	Fecha Final	Comentario
Alvaro Orozco Pérez	1.0	26/Ene/2024	19/Mayo/2024	Creación de laplataforma

Este documento plasma las necesidades, deseos, objetivos, metas y alcances que el cliente espera como resultado de la ejecución del proyecto. Se establecen las características
generales del proyecto y funciona como la señal formal de inicio de proyecto.

1.0.0 Generales

Nombre del proyecto

EduShare (Plataforma Para Compartir Información De Forma Eficiente Y Atinada)Área y persona solicitante
Alumnos y personas interesados
en ciertos temas Fecha de
autorización del proyecto
26/Ene/2024

Administrador del proyecto Asignado

Alvaro Orozco Pérez. Quien tendrá el nivel total de autoridad sobre el proyecto y deberá informar de los avances del mismo subdirector académico.

1.A.1 Antecedentes

Se propone la implementación y expansión del proyecto EduShare para

abordar las
necesidades actuales en el ámbito académico. En la actualidad, el proceso de intercambioy acceso a recursos educativos se realiza de manera limitada, con métodos tradicionales que no aprovechan las ventajas de la tecnología. Con la creación de EduShare, se busca transformar y mejorar la experiencia educativa de los estudiantes universitarios.

Actualmente, los estudiantes se enfrentan a obstáculos para compartir y acceder a
recursos educativos de manera eficiente. EduShare pretende superar estas limitaciones,

proporcionando una plataforma en línea donde los estudiantes pueden colaborar de manera efectiva al compartir apuntes, presentaciones, y otros materiales de estudio. Esteproyecto tiene como objetivo principal fomentar la colaboración y mejorar la productividad académica, permitiendo a los estudiantes acceder fácilmente a recursos dealta calidad.

1. A.2 Objetivo del proyecto y metas

1. Ampliación de Funcionalidades:
    - Desarrollar nuevas funciones para permitir a los estudiantes cargar, buscar y comentar sobre una variedad de recursos educativos, incluyendo apuntes, presentaciones, y material de estudio.
2. Interfaz Intuitiva:
    - Diseñar una interfaz de usuario intuitiva y fácil de usar, que promueva unaexperiencia fluida para los usuarios, independientemente de su nivel de habilidad tecnológica.
3. Colaboración Efectiva:
    - Facilitar la colaboración entre estudiantes al implementar herramientas que fomenten la interacción, como comentarios y opciones de búsqueda.
4. Optimización del Tiempo:
    - Reducir el tiempo necesario para realizar procesos académicos, como la revisión frente a grupo, permitiendo a los profesores y estudiantes utilizarla plataforma de manera eficiente.
5. Comunidad Educativa:
    - Establecer una comunidad en línea dentro de EduShare que promueva la participación, el intercambio de conocimientos y la construcción de una redacadémica sólida.
6. Seguridad y Privacidad:
    - Implementar medidas de seguridad para proteger la información de los usuarios y los recursos cargados en la plataforma, cumpliendo con
      estándares de privacidad.
7. Escalabilidad:

- Diseñar la plataforma para manejar un crecimiento escalonado de usuarios,anticipando una mayor adopción tanto a corto como a largo plazo.

8. Mantenimiento y Actualizaciones:

- Establecer un plan de mantenimiento regular y proporcionar actualizaciones periódicas para garantizar que la plataforma siga siendorelevante y eficiente con el tiempo.

1.A.3 Alcance del proyecto

No.	Entregables/Productos
1	Módulo de login
2	Interfaz para el usuario
3	Registro para nuevos usuarios
4	Modulo para subir archivos
5	Interfaz para administrador
6	Interfaz para moderador
7	Sistema de búsqueda de archivos
8	Etiqueta de todos los archivos subidos

1. A.5 Identificación de Interesados

Requerimientos de Aceptación	
Entregables obtenidos a la fecha	Descripción de requerimientos de aceptación

Módulo de login	-Espacio para ingresar los datos a realizar -Validar si los datos existen tal cal losescribió -Interfaz y que se vea estético

Interfaz para el usuario	-Que se vea bonito y estético -Que se muy intuitiva y se pueda familiarizarfácilmente -Que tengo colores que hagan buen contraste
Registro para nuevos usuarios	-Podrá registrarse cualquier usuario -Que este a solo unos clics de la página depublicidad -Que los datos que ingresen sean seguros
Modulo para subir archivos	-Un botón para seleccionar el archivo ysubirlo -Que se puedan subir varios tipos dearchivos -
Interfaz para administrador	-Búsqueda para usuarios -Podrá eliminar, modificar, agregar o mostrar cualquier usuario -Podrá hacer lo que cualquier usuario común
Interfaz para moderador	-Seleccionara que se puede subir y que no lopuede hacer -podrá hacer lo que un usuario común -Tendrá acceso a contenido de todos losusuarios -podrá eliminar, modificar, agregar o mostrar el contenido de cualquier usuario
Sistema de búsqueda de archivos	-Podrá buscar cualquier cosa de forma eficiente -búsqueda con expresiones regulares -búsqueda con etiquetas
Etiqueta de todos los archivos subidos	-Podrás ver de qué se trata sin abrir el archivo -Ayuda a las búsquedas mas eficientes

1.A.5 Identificación de interesados

Áreas y personal involucrado en el Proyecto		
Áreas	Participante	Responsabilidad

Moderador	Personal que se le contratara para estafunción	Verificar que el contenido que se suba sea apto
Administrador	Desarrollador del software	Sancionar en caso de incumplimientos de normas
Usuario	Cualquier persona que esteinteresado en utilizar la plataforma	Subir, descargar, mostrar, eliminar, modificar, comentar y chats con otros usuarios

1. A.6 Riesgos de Alto Nivel del Proyecto

Descripción del Riesgo	Posibles Respuestas
1. Brechas en la Seguridad deDatos:	- Implementar protocolos de seguridad. - Utilizar encriptación y hash para datos sensibles. - Realizar auditorías de seguridad regulares.
2. Fallos en la Autenticación:	- Aplicar prácticas seguras de almacenamiento y manejo de contraseñas.
3. Problemas de Escalabilidad:	- Diseñar la arquitectura de Edushare con escalabilidaden mente. - Monitorear el rendimiento y anticipar aumentos de carga.
4. Interrupciones en el Servicio:	- Establecer un plan de contingencia para mantener servicios críticos. - Implementar redundancias y backups para recuperación de datos. - Monitorear la disponibilidad constantemente.
5. Ataques de Inyección de Datos:	- Validar y sanitizar todas las entradas de usuario para prevenir inyecciones.

6. Desactualización de Tecnologías:	- Establecer un plan de mantenimiento regular para actualizar bibliotecas y componentes. - Monitorear las vulnerabilidades y parches de seguridad disponibles. - Evaluar nuevas tecnologías que puedan mejorar laseguridad.
7. Falta de Adopción por Parte de Usuarios:	- Realizar sesiones de capacitación y demostraciones para familiarizar a los usuarios con Edushare

	- Ofrecer soporte técnico y documentación detallada.
8. Problemas de Colaboración entre Áreas Involucradas:	- Facilitar la comunicación a través de reuniones regulares y herramientas de colaboración. - Establecer protocolos claros de responsabilidades yprocesos. - Implementar un sistema de gestión de proyectos para un seguimiento más efectivo.

1. A.7 Supuestos y Fuera de Alcance

Supuestos Generales
• Se asume que los usuarios participarán de manera voluntaria en la plataforma, contribuyendo y beneficiándose del intercambio de recursos educativos. • Se asume que los usuarios proporcionarán información precisa y auténtica al cargarrecursos educativos, y que el sistema de reporte ayudará a mantener la integridad del contenido. • Se presupone que los usuarios respetarán los derechos de autor y la propiedad intelectual al compartir recursos, y que la plataforma no se utilizará para infringirderechos legales. • Se asume que los usuarios tendrán acceso a una conexión a Internet para utilizar laplataforma de manera efectiva. • Se presupone que los usuarios interactuarán de manera constructiva, proporcionando calificaciones y comentarios útiles para mejorar la calidad de losrecursos educativos. • Se asume que la plataforma implementará medidas de seguridad para proteger lainformación personal de los usuarios y prevenir cualquier intento de acceso no autorizado. • Se presupone que la plataforma contendrá una variedad de recursos educativos que abarcan diferentes áreas de estudio, niveles académicos y formatos de archivo. • Se presupone que los usuarios están familiarizados con la tecnología necesaria

para utilizar la plataforma, incluida la subida y descarga de archivos, así como lainteracción con la interfaz en línea.

Fuera del Alcance

- La detección y gestión de contenido ilegal o inapropiado puede ser un desafío complejo y, por lo tanto, podría considerarse fuera del alcance del proyecto. La

responsabilidad de moderar y eliminar dicho contenido podría recaer en otras entidades.

- La plataforma EduShare puede proporcionar recursos educativos, pero no debe considerarse como un sustituto de asesoramiento académico profesional. La toma de decisiones académicas importantes debe ser supervisada por profesionales de la educación.
- EduShare puede facilitar el acceso a recursos educativos, pero no puede garantizar el éxito académico de los estudiantes. Factores externos, habilidades individuales y esfuerzo personal también desempeñan un papel crucial en el rendimiento académico.
- La integración directa con sistemas universitarios, como la gestión de cursos, calificaciones y horarios, puede considerarse fuera del alcance inicial del proyecto debido a la complejidad y posibles restricciones tecnológicas.
- La creación de contenido educativo propio por parte de la plataforma, como cursos en línea, tutoriales o material de estudio exclusivo, podría considerarse fuera del alcance inicial y requeriría recursos adicionales y expertise específica.
- La traducción automática de recursos educativos a múltiples idiomas podría ser un desafío técnico considerable y podría considerarse fuera del alcance inicial.
- Estrategias de publicidad y monetización directa, como la incorporación de anuncios publicitarios en la plataforma, podrían considerarse fuera del alcance inicial, dependiendo de los objetivos y valores del proyecto.
- La plataforma puede proporcionar recursos, pero no se encargará de ofrecer formación académica específica o personalizada. Esto queda en manos de las instituciones educativas y profesionales de la educación.

Manuales de Usuario

Manuales de Usuario en Drive (Carpeta con los dos manuales)

https://drive.google.com/drive/folders/1S3FihO13AVYa9nFN5xCgYxMB94Deb9Rv?usp=sharing

Administrador
https://edushare.top/resources/Archivos/Manual%20de%20Usuario%20-%20Administrador.pdf

Usuario
https://edushare.top/resources/Archivos/Manual%20de%20Usuario.pdf

Costos

Indirectos						
Actividad	Precio	Periodo	Costo/Semana	Importe	Cantidad Proyectos	Total
Renta	2300	Mes	575	8050	1	8050
Agua	200	Mes	50	700	1	700
Limpieza	1600	Mes	400	5600	1	5600
Otros						
Internet	450	Mes	112.5	1575	1	1575
Electricidad	150					
						15925

Directos						
Actividad	Precio	Periodo	Costo/Semana	Importe	Cantidad Proyectos	Total
Certificado	2500	Anual	865.38	12115.38	1	12115.38
Calidad	8000	Mes	2000.00	28000.00	1	28000.00
Mano de obra	7500	Semanal	7500.00	105000.00	1	105000.00
LapTop				0	1	0.00
Licencias				0.00	1	0.00
						145115.38

Costo	Año	Devaluacion	Real	Div 52 Sem	14 Sem
16200	1	4050	12150	77.88	1090.38462
	2	3037.5	9112.5	58.41	817.788462
	3	2278.125	6834.375	43.81	613.341346
	4	1708.59375	5125.78125	32.86	460.00601
	5	1281.44531	3844.33594	24.64	345.004507
	6	961.083984	2883.25195	18.48	258.75338

Suma Total	$161,858.17

Puntos de Fusion

Entradas

Núm.	Entrada	Archivos referidos	Elementos de	Nivel	Valor
1	AgregarPersonas	1	3	Bajo	Alto
2	AgregarEstudiantes	2	8	Medio	4
3	AgregarAdministradores	2	5	Medio	4
4	AgregarChats	2	7	Medio	4
5	AgregarComentario_	2	9	Medio	4
6	AgregarEtiquetas	2	7	Medio	4
7	AgregarModerador_	2	5	Medio	4
8	AgregarPublicaciones	1	4	Bajo	3
9	AgregarRecursos	2	9	Medio	4
10	AgregarReportes	2	8	Medio	4
11	AgregarReportesAceptados	2	10	Medio	4
12	EditarPersonas	1	3	Bajo	3
13	EditarEstudiantes	2	8	Medio	4
14	EditarAdministradores	2	5	Medio	4
15	EditarChats	2	7	Medio	4
16	EditarComentarios	2	9	Medio	4
17	EditarEtiquetas	2	7	Medio	4
18	EditarModeradores	2	5	Medio	4
19	EditarPublicacione_	1	4	Bajo	3
20	EditarRecursos	2	9	Medio	4
21	EditarReportes	2	8	Medio	4
22	EditarReportesAceptados	2	8	Medio	4
23	EliminarPersonas	1	3	Bajo	3
24	EliminarEstudiante_	2	8	Medio	4
25	EliminarAdministradores	2	5	Medio	4
26	EliminarChats	2	7	Medio	4
27	EliminarComentari_	2	9	Medio	4
28	EliminarEtiquetas	2	7	Medio	4
29	EliminarModerador_	2	5	Medio	4
30	EliminarPublicacio_	1	4	Bajo	3
31	EliminarRecursos	2	9	Medio	4
32	EliminarReportes	2	8	Medio	4
33	EliminarReportesAceptados	2	9	Medio	4

Imagen 108

Subtotal Entradas	123

Imagen 109

Salidas

Núm.	Salida	Archivos referidos	Elementos de datos	Nivel	Valor
1	Vista Usuarios	3	14	Alto	7
2	Estudiantes	4	8	Bajo	4
3	Publicaciones	5	11	Bajo	4
4	Vista Reportes	5	8	Alto	7
5	Chats	2	1	Bajo	4
6	ChatGrupal	3	4	Bajo	4
7	ModeradoresActivos	2	4	Bajo	4
8	Comentarios	3	4	Bajo	4
9	Reportes Aceptados	5	4	Medio	5
10	Reportes Rechazados	5	4	Medio	5

Imagen 110

Subtotal Salidas	48

Imagen 111

Peticiones

Núm.	Petición	Archivos referidos	Elementos de datos	Nivel	Valor
1)	Datos de usuario	3	6	Bajo	3
2)	Lista de Usuarios	4	7	Medio	4
3)	Lista Estudiantes	2	5	Bajo	3
4)	Lista publicaciones	3	4	Bajo	3
5)	Lista comentarios	3	6	Bajo	3
6)	Chats	3	4	Medio	4

Imagen 112

Subtotal Peticiones	20

Imagen 113

Núm.	Archivo	Elementos de registro	Elementos de datos	Nivel	Valor
1	Personas	11	26	Alto	15
2	Estudiante	11	26	Alto	15
3	Administrador	11	26	Alto	15
4	Moderador	11	26	Alto	15
5	Etiquetas	11	26	Alto	15
6	Recursos	11	26	Alto	15
7	Publicaciones	11	26	Alto	15
8	Reportes	11	26	Alto	15
9	Reportes Aceptados	11	26	Alto	15
10	Comentarios	11	26	Alto	15
11	Chats	11	26	Alto	15

Imagen 114

Subtotal Archivos internos	165

Imagen 115

Núm.	Archivo	Elementos de registro	Elementos de datos	Nivel	Valor
1)	No Tengo Archivos Externos				

Imagen 116

Subtotal Archivos externos	0

Imagen 117

0	1	2	3	4	5
Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial

1.	¿Requiere el sistema copias de seguridad y de recuperación fiables?	**5**
2.	¿Se requieren comunicaciones de datos?	**5**
3.	¿Existen funciones de procesamiento distribuido?	**0**
4.	¿Es crítico el rendimiento?	**5**
5.	¿Será ejecutado el sistema en un entorno operativo existente y fuertemente utilizado?	**5**
6.	¿Requiere el sistema entrada de datos interactiva?	**5**
7.	¿Requiere entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas o variadas operaciones ?	**5**
8.	¿Se actualizan los archivos maestros de forma interactiva?	**5**
9.	¿Son complejas las entradas, las salidas, los archivos o las peticiones?	**3**
10.	¿Es complejo el procesamiento interno?	**3**
11.	¿Se ha diseñado el código para ser reutilizable?	**3**
12.	¿Están incluidas en el diseño la conversión y la instalación?	**3**
13.	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	**2**
14.	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	**5**

Imagen 118

Total	54

Imagen 119

Entrada	123
Salida	48
Peticiones	20
Archivos Internos	210
Archivos Externos	0
Total no ajustados	401
Total Ajustados	54
Pf	477.19

Imagen 120

Certificaciones

No contamos con ninguna certificación como desarrolladores, sin embargo, cumple las normas de seguridad básicas para el uso y la protección de datos

Riesgos

Magnitud de Daño:	
1	Nulo
2	Bajo
3	Mediano
4	Alto

Llenar los espacios necesarios de acuerdo a la clasificacion de los riesgos. En cada uno de los campos de la tabla se aplica de acuerdo al ejemplo de llenado:

Clasificacion	Riesgo	Magnitud de Daño: (1 = Nulo 2 = Bajo 3=Mediano 4 = Alto)	Observaciones	Tiempo respuesta	Responsable	Actividades a realizar	Resultados posibles	Costo por mes
Riesgos Técnicos	Falla del Servidor	4	Caída del servidor de alojamiento	2 días	Equipo de Desarrollo	Contactar al proveedor de alojamiento, migrar a un servidor alternativo si es necesario.	Tiempo de inactividad del sitio, pérdida de datos.	800
Riesgos Operativos	Falta de Comunicación	2	Comunicación ineficaz entre el equipo	Continuo	Gerente de Proyecto	Establecer canales de comunicación claros, reuniones regulares.	Malentendidos, falta de alineación.	500
Riesgos Externos	Cambios en las Regulaciones	3	Cambios en las leyes de privacidad de datos	15 días	Equipo Legal	Monitorear cambios regulatorios, ajustar la aplicación según sea necesario.	Multas por incumplimiento, pérdida de confianza del cliente.	900
Riesgos de Recursos Humanos	Problemas de Capacitación	2	Falta de habilidades técnicas en el equipo	10 días	Gerente de Proyecto	Ofrecer capacitación, contratar personal con las habilidades necesarias.	Retrasos en el proyecto, baja calidad del trabajo.	800
Riesgos de Recursos Humanos	Problemas de Capacitación	2	Falta de habilidades técnicas en el equipo	10 días	Gerente de Proyecto	capacitación, contratar personal con las habilidades necesarias.	Retrasos en el proyecto, baja calidad del trabajo.	800
Riesgos de Mercado	Cambios en las Tendencias de Usuario	4	Cambios en las preferencias de los usuarios o en las tendencias de las redes sociales que afectan la demanda de la red social.	Continuo	Equipo de Marketing	Realizar estudios de mercado frecuentes, adaptar la estrategia de marketing según sea necesario.	Pérdida de usuarios, disminución de ingresos.	700
Riesgos Tecnológicos	Fallas en la Seguridad de la Aplicación	5	Vulnerabilidades de seguridad en la aplicación que podrían ser explotadas por atacantes.	3 días	Equipo de Seguridad Informática	Realizar auditorías de seguridad regulares, parchear y corregir vulnerabilidades identificadas.	Brechas de seguridad, robo de datos sensibles.	1000
Riesgos Operativos	Problemas de Escalabilidad del Servidor	4	Incapacidad del servidor para manejar un aumento repentino en la carga de usuarios.	5 días	Equipo de Desarrollo	Implementar soluciones de escalabilidad como el uso de servidores en la nube o la optimización del código.	Rendimiento lento del sitio, caídas del servidor.	900
Riesgos Legales	Litigios por Derechos de Autor o Propiedad Intelectual	4	Demandas legales relacionadas con la violación de derechos de autor, marcas comerciales u otros derechos de propiedad intelectual.	Variable	Equipo Legal	Consultar con abogados especializados, revisar y actualizar políticas de uso y derechos de autor.	Multas, pérdida de reputación, cierre del proyecto.	1200

Cronograma


Actividades
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
20 Ene-02 Feb
05 Feb-09 Feb
12 Feb-16 Feb
19 Feb-23 Feb
26 Feb-01 Mar
04 Mar-08Mar
11 Mar-15 Mar
18 Mar-22 Mar
25 Mar-29 Mar
01 Abril-05 Mar
08 Abril-12 Abr
15 Abril-19 Abr
22 Abril-26 Abr
29 Abril-03 Mar
06 May-10 May
13 May-17 May
20 May-24 may
Requerimientos
Para entrevistar al cliente y comprender sus necesidades en un proyecto de software, escucha atentamente, haz preguntas claves y toma notas
Entrevista
Contrato
Diagramas
Cotizacion (alcance, tiempo, costos)
Planeación y desarrollo
Diseño de la base de datos
Desarrollo de Front-End
Desarrollo de Back-End
Pruebas
Caja blanca
Caja Negra
Sobre carga
Implementación
Entrega del sistema
Documentación