



IES San Juan de la Cruz



Unión Europea
Fondo Social Europeo
"El FSE invierte en tu futuro"

TÉCNICO SUPERIOR EN DESARROLLO DE APLICACIONES WEB (IFCS03)

MÓDULO: PROYECTO. CURSO: 2024-2025

CONVOCATORIA: ORDINARIA

Título del proyecto: TASK-CRAFT

PROFESOR TUTOR: Mercedes Limón Echevarría

Alumnos:

APELLIDOS Y NOMBRE: MOLINA ARELLANO NEIL	DNI: 54444173E
---	-----------------------

MODALIDAD DE PROYECTO: Proyectos de gestión

ÍNDICE

1. JUSTIFICACIÓN.....	3
2. INTRODUCCIÓN.....	4
3. TOMA DE REQUISITOS.....	5
4. OBJETIVOS.....	36
5. DESARROLLO.....	37
5.1. METODOLOGÍA.....	37
5.2. TECNOLOGÍAS.....	45
5.3. PROTOTIPADO Y USABILIDAD.....	46
5.4. MODELADO DE DATOS.....	64
5.5. SEGURIDAD Y AUTENTICACIÓN.....	71
5.6. DISEÑO E IMPLEMENTACIÓN.....	72
5.7. PRUEBAS Y CALIDAD DEL SOFTWARE.....	74
6. DESPLIEGUE DEL PRODUCTO.....	78
7. PRESUPUESTO, RIESGOS, VIABILIDAD.....	80
8. DIFICULTADES ENCONTRADAS Y RESOLUCIÓN.....	82
9. CONCLUSIONES.....	83
10. REFERENCIAS BIBLIOGRÁFICAS.....	84
11. ANEXOS.....	85

1. JUSTIFICACIÓN

En un mundo cada vez más ocupado y lleno de distracciones, resulta fundamental contar con una herramienta que facilite la organización de las actividades diarias de manera eficiente y efectiva. La falta de organización puede conllevar fácilmente a la pérdida de tiempo en actividades poco productivas, desviándonos de nuestros objetivos y metas personales. Esta necesidad surge en situaciones donde se busca equilibrar la adquisición de nuevas habilidades, como aprender a programar tanto de forma académica como autodidacta, con otras actividades recreativas o sociales, como asistir al gimnasio o pasar tiempo con amigos. La inspiración proviene de obras como "Hábitos Atómicos" de James Clear, donde se destaca la importancia de desarrollar hábitos pequeños pero consistentes en el día a día para alcanzar metas personales y convertirse en la mejor versión de uno mismo. En este contexto, surge la idea de crear una aplicación que facilite la creación y seguimiento de nuevos hábitos, proporcionando una herramienta de organización más efectiva para el manejo de múltiples actividades. (Clear, 2020)

2. INTRODUCCIÓN

En un contexto contemporáneo marcado por la velocidad y la constante presencia de distracciones, surge la necesidad imperante de gestionar eficazmente el tiempo y las responsabilidades. Este proyecto de fin de grado aborda esta premisa mediante la creación de una aplicación de gestión de tareas y metas, con el propósito de mejorar la productividad y la satisfacción del usuario al cumplir con sus compromisos diarios y alcanzar objetivos a largo plazo.

El objetivo de esta investigación consiste en desarrollar una herramienta intuitiva y eficiente que facilite a los usuarios la organización de sus actividades cotidianas y la planificación de sus metas de manera efectiva. La motivación principal detrás de esta indagación radica en la creciente demanda de soluciones que permitan a las personas manejar sus responsabilidades de forma más efectiva, aumentando así su eficiencia personal.



Para lograr dicho propósito, se realizó una revisión exhaustiva de estudios sobre gestión del tiempo y productividad, con el fin de identificar los principales obstáculos y desafíos que los usuarios enfrentan en su organización diaria. El desarrollo de la aplicación se llevó a cabo empleando un enfoque centrado en el usuario, lo cual incluyó el diseño de una interfaz intuitiva y la implementación de mecanismos de seguimiento y organización de tareas.

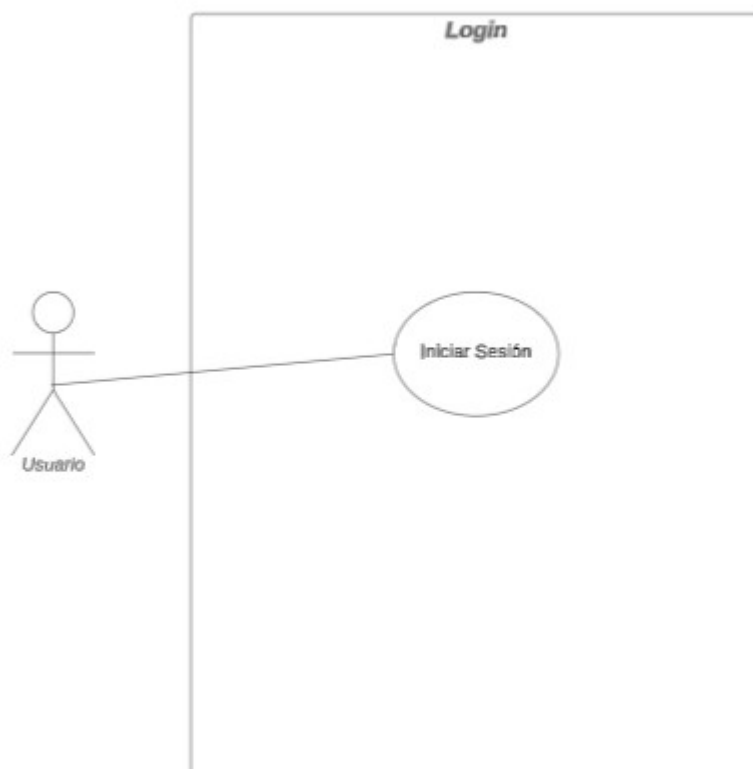
Los precedentes y prácticas en el ámbito de la gestión del tiempo demuestran que una planificación adecuada y el uso de herramientas especializadas pueden incidir significativamente en la eficiencia personal. Con los avances tecnológicos y el diseño de aplicaciones, surgen nuevas oportunidades para desarrollar herramientas más intuitivas y eficaces.

Este proyecto proporciona una solución práctica para la gestión de tareas y metas, con el propósito de ayudar a las personas a optimizar su tiempo y alcanzar sus objetivos de manera efectiva. El proceso de desarrollo de la aplicación garantiza que esta herramienta responda a las necesidades actuales de los usuarios.

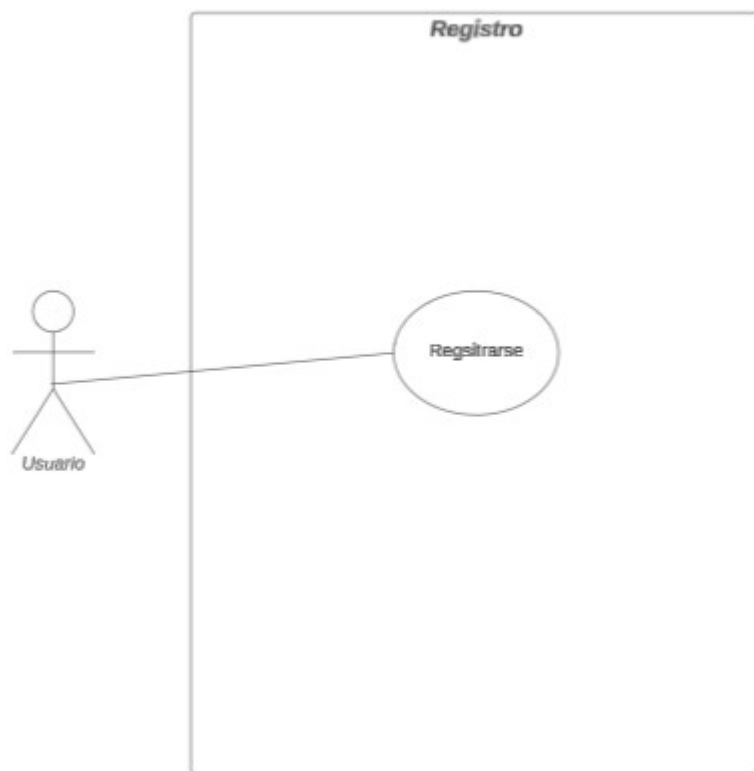
3. TOMA DE REQUISITOS

1. Que el usuario pueda **iniciar sesión**: Mediante ingresando unas credenciales (correo electrónico y contraseña), el usuario pueda acceder de manera segura a las funcionalidades de la aplicación. Al autenticarse correctamente, el sistema generará una sesión segura o token con JWT.
2. El programa tendrá la opción de ofrecer al usuario un formulario de **registro**, el cual el cliente podrá crear una credenciales personalizadas, con la **contraseña encriptada** para ofrecer mayor seguridad a la aplicación.
3. **Gestión de diarios**: se podrá **crear, editar, eliminar o visualizar** entradas de **diarios**, asociadas a un usuario. Cada uno de estos tendrá la fecha, el título y un contenido.
4. **Gestión de tareas**: se podrá **crear, editar, eliminar o visualizar** (tanto en un calendario como en una página aparte) **tareas**, asociadas a un usuario. Cada uno de estos contendrá el título, una breve descripción, un estado cargado desde la base de datos y una categoría, el cual podrá ser personalizada por el usuario o cargada desde la base de datos.
5. **Gestión de usuario**: el usuario podrá **cambiar sus credenciales, cerrar sesión o eliminar** su cuenta.
6. **La aplicación debe ser web**: debe de tener una interfaz amigable, accesible y responsiva que se pueda manejar desde un navegador.
7. **Seguridad**: la aplicación tendrá las **contraseñas encriptadas** y las sesiones estarán protegidas con **JWT**, guardadas en una **cookie**.
8. **Gestión de amigos**: el usuario podrá añadir a amigos mediante solicitudes de amistad. También se podrán compartir tareas entre ellos.

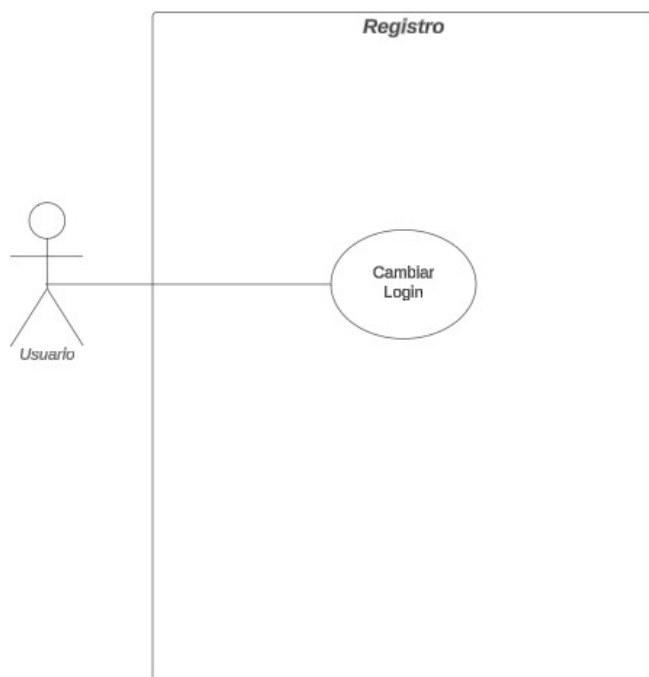
CASOS DE USO



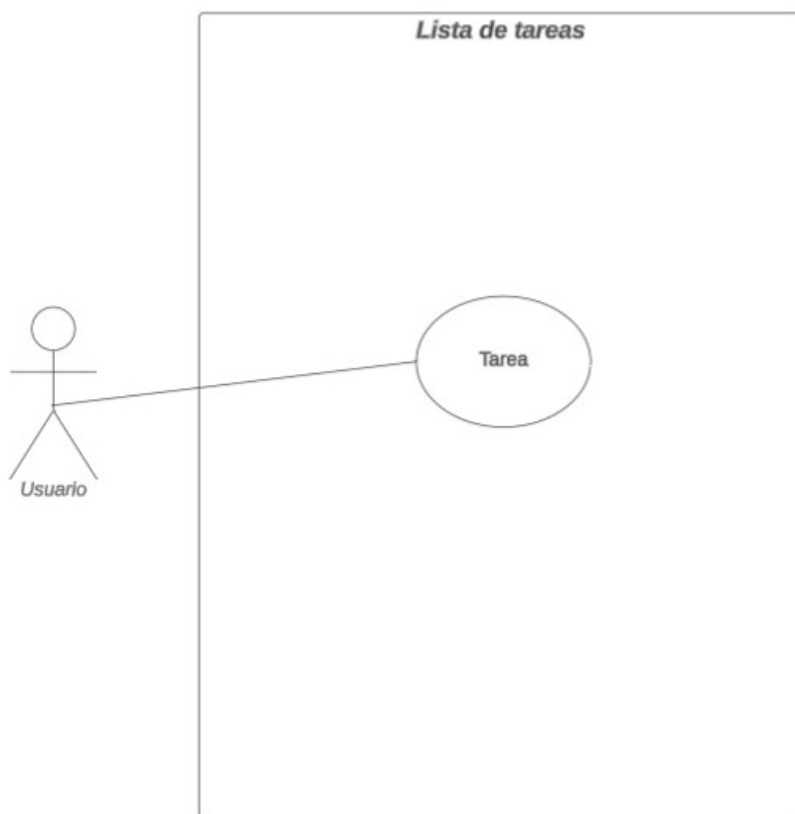
Nombre	CU01
Actor	Usuario
Descripción	El usuario inicia sesión
Evento desencadenado	La pantalla se cambia a la de la aplicación principal si las credenciales son correctas.
Pasos realizados	<ol style="list-style-type: none">1. Pulsar el botón de cambio de pantalla2. Se cambia de pantalla de Registro
Precondiciones	Abrir la pantalla de Login
Postcondiciones	Cambiar de pantalla de Dashboard
Vista	Login



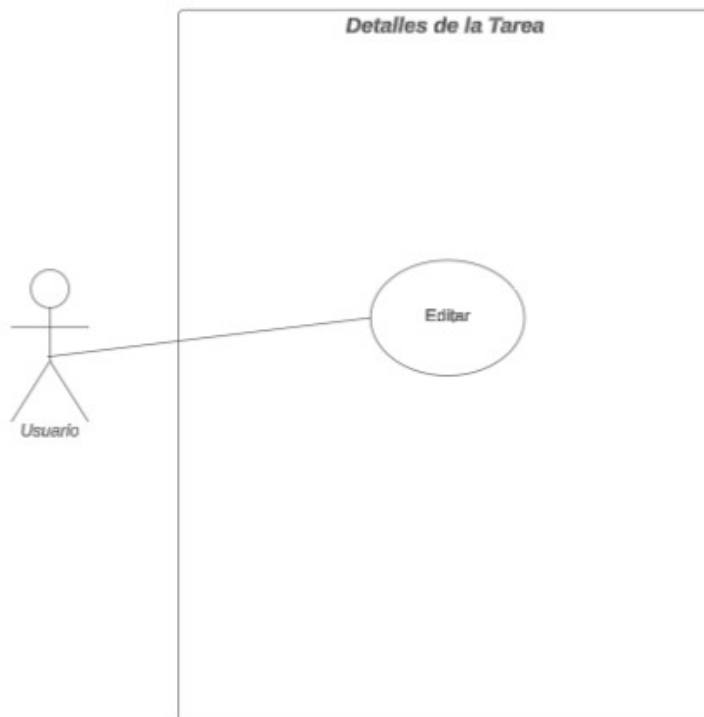
Nombre	CU02
Actor	Usuario
Descripción	El usuario se registra
Evento desencadenado	La pantalla se cambia a la de Login
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar el botón de cambio de pantalla 2. Se cambia de pantalla de Registro
Precondiciones	Abrir la pantalla de Registro
Postcondiciones	Cambiar de pantalla de Dashboard
Vista	Register



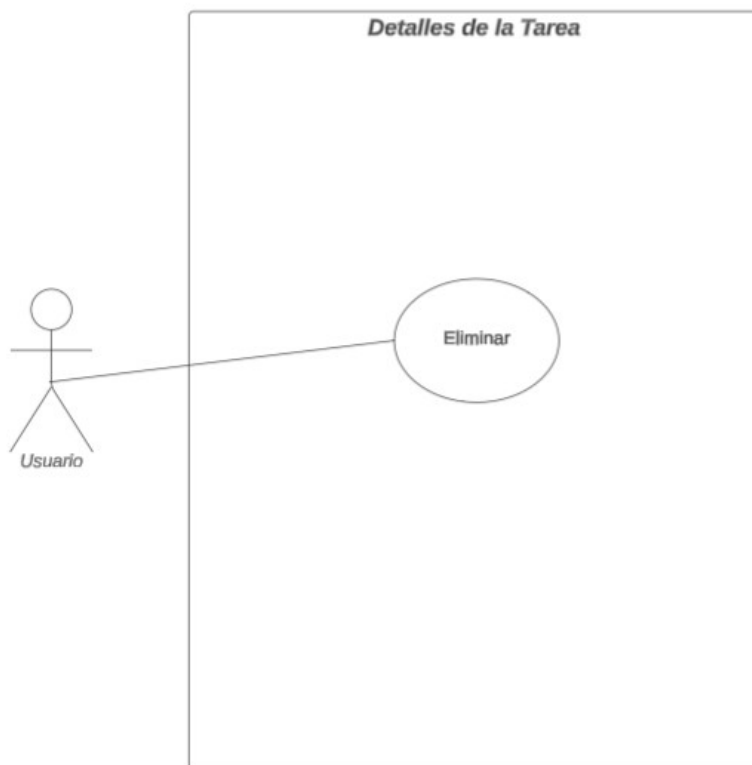
Nombre	CU03
Actor	Usuario
Descripción	El usuario se cambia de pantalla a la de Register
Evento desencadenado	La pantalla se cambia a la de Register
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar el botón de cambio de pantalla 2. Se cambia de pantalla de Registro
Precondiciones	Abrir la pantalla de Registro
Postcondiciones	Cambiar de pantalla de Dashboard
Vista	Register



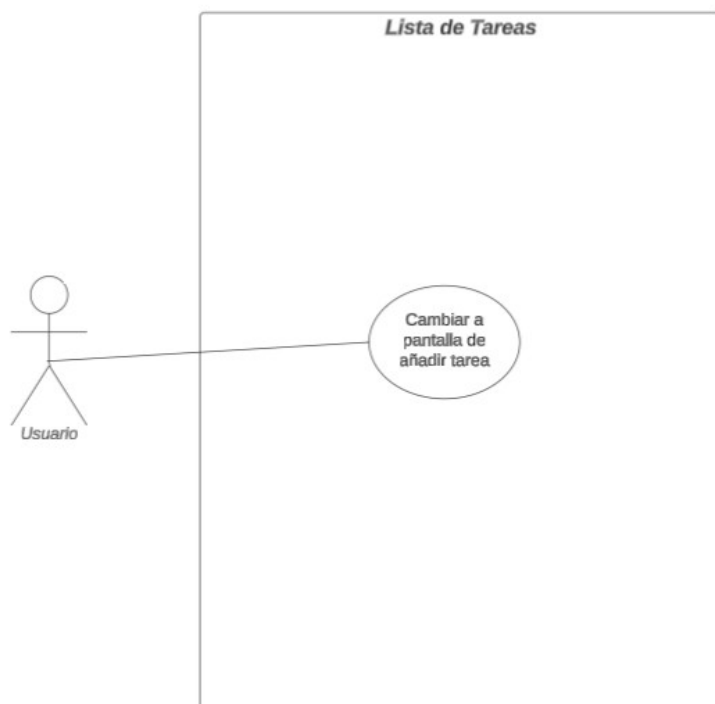
Nombre	CU04
Actor	Usuario
Descripción	El usuario se cambia de pantalla a la de Detalles Tarea
Evento desencadenado	La pantalla se cambia a la de Detalles Tareas
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar la tarea 2. Se cambia de pantalla de Detalles Tareas
Precondiciones	Abrir la pantalla de Lista de Tareas
Postcondiciones	Cambiar de pantalla de Detalles Tareas
Vista	Lista de Tareas



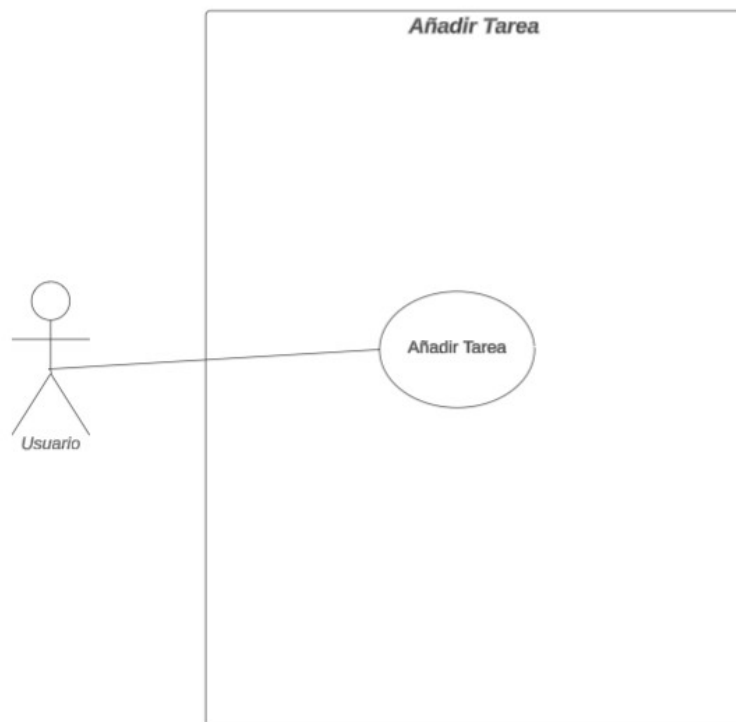
Nombre	CU05
Actor	Usuario
Descripción	El usuario edita la tarea en la pantalla de Detalles Tarea
Evento desencadenado	La pantalla se cambia a la de Lista de Tareas, se cambia los datos de la tarea tanto en la UI como en la base de datos
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar la tarea 2. Se cambian los datos de la tarea 3. Se cambia de pantalla a la Lista de Tareas
Precondiciones	Abrir la Tarea
Postcondiciones	Cambiar de pantalla de Lista de Tareas
Vista	Detalles Tarea



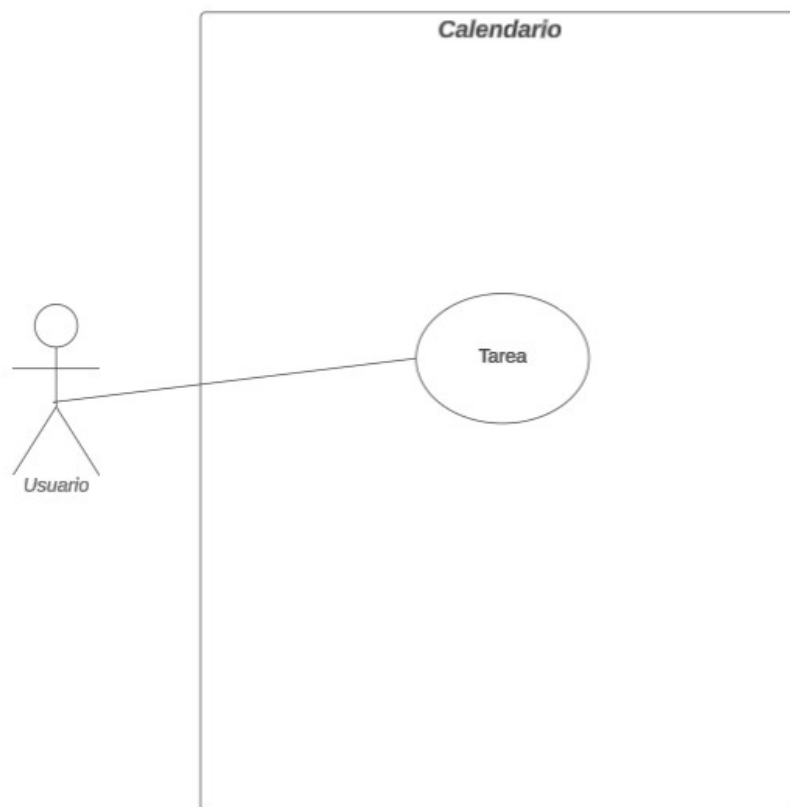
Nombre	CU06
Actor	Usuario
Descripción	El usuario se elimina la tarea en la pantalla de Detalles Tarea
Evento desencadenado	La pantalla se cambia a la de Lista Tareas y se elimina la tarea tanto en la UI como en la base de datos
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar la tarea 2. Se elimina la tarea 3. Se cambia de pantalla de Detalles Tareas
Precondiciones	Abrir la Tarea
Postcondiciones	Cambiar de pantalla de Lista de Tareas
Vista	Detalles Tarea



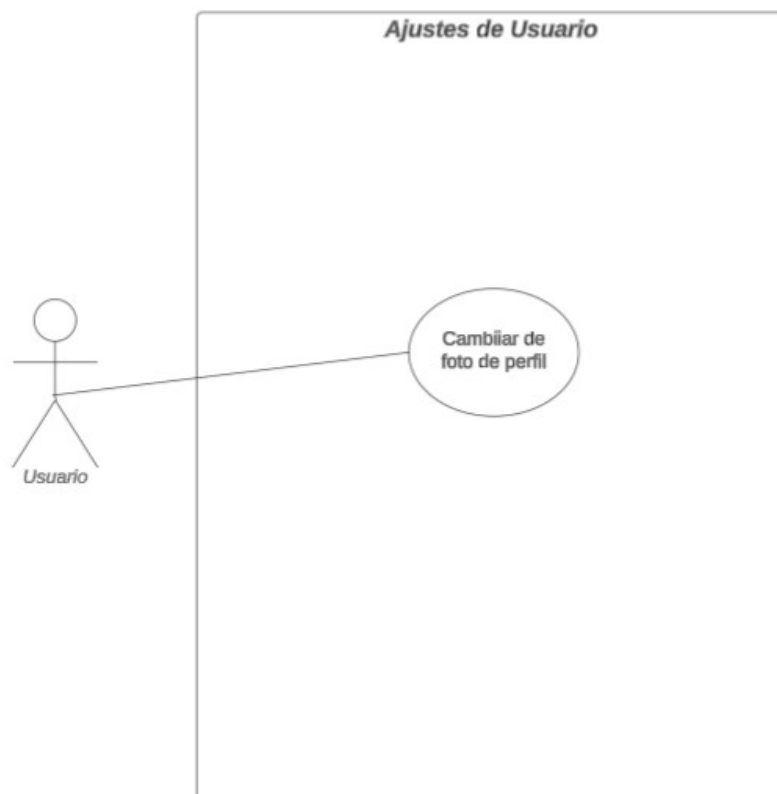
Nombre	CU07
Actor	Usuario
Descripción	El usuario pulsa el botón para cambiar a la pantalla de Añadir Tareas
Evento desencadenado	La pantalla se cambia a la de Añadir Tareas
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar el botón 2. Se cambia de pantalla de Añadir Tareas
Precondiciones	Estar en la pantalla de Lista de Tareas
Postcondiciones	Cambiar de pantalla de Añadir Tareas
Vista	Lista de Tareas



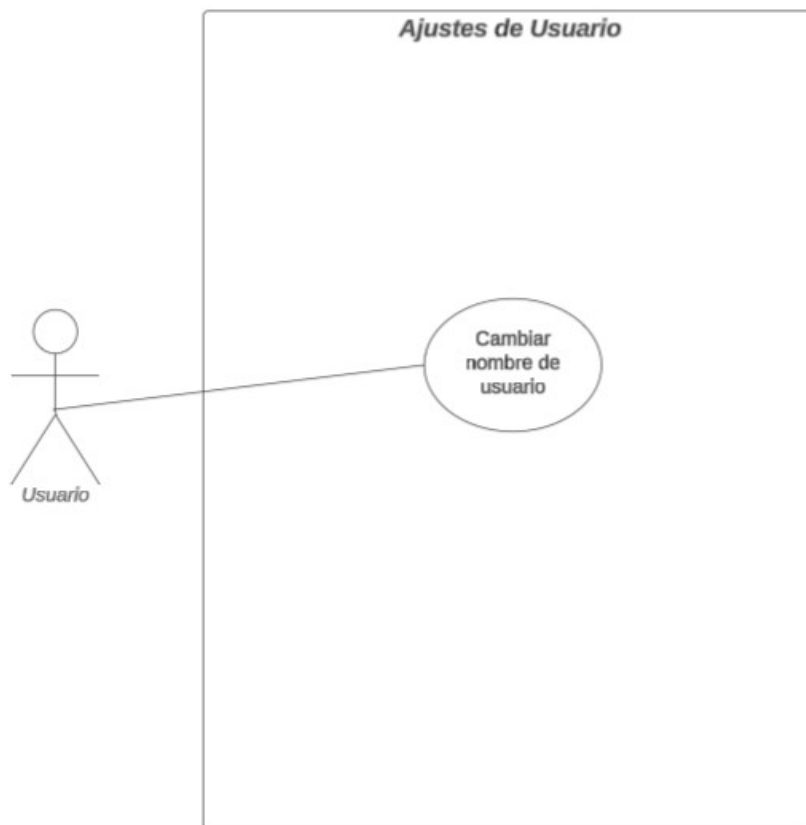
Nombre	CU08
Actor	Usuario
Descripción	El usuario se cambia la pantalla a la de Detalles Tarea
Evento desencadenado	La pantalla se cambia a la de Añadir Tareas
Pasos realizados	1. Pulsar el botón 2. Se cambia de pantalla de Añadir Tareas
Precondiciones	Estar en la pantalla de Lista de Tareas
Postcondiciones	Cambiar de pantalla de Añadir Tareas
Vista	Añadir de Tareas



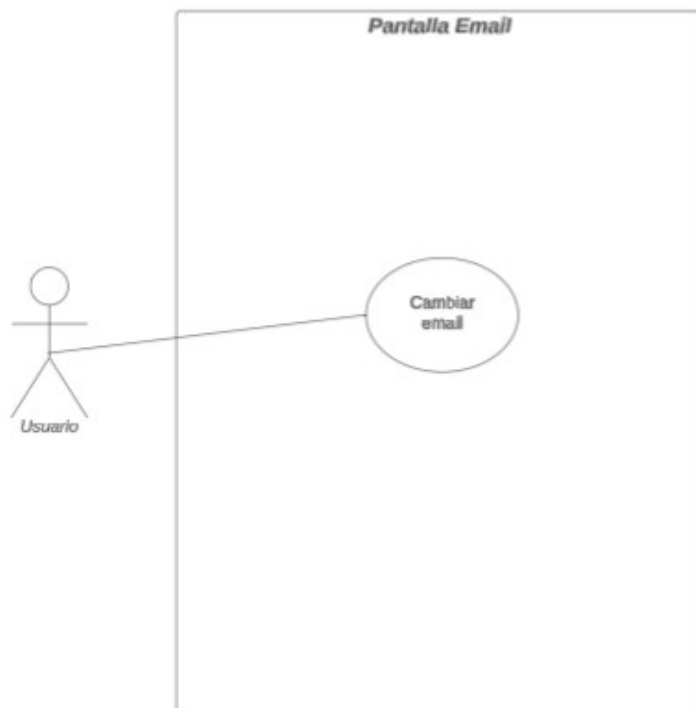
Nombre	CU09
Actor	Usuario
Descripción	El usuario se cambia de pantalla a la de Detalles Tarea
Evento desencadenado	La pantalla se cambia a la de Detalles Tareas
Pasos realizados	<ol style="list-style-type: none">1. Pulsar la tarea o el evento en el calendario2. Se cambia de pantalla de Detalles Tareas
Precondiciones	Abrir la pantalla del Calendario
Postcondiciones	Cambiar de pantalla de Detalles Tareas
Vista	Calendario



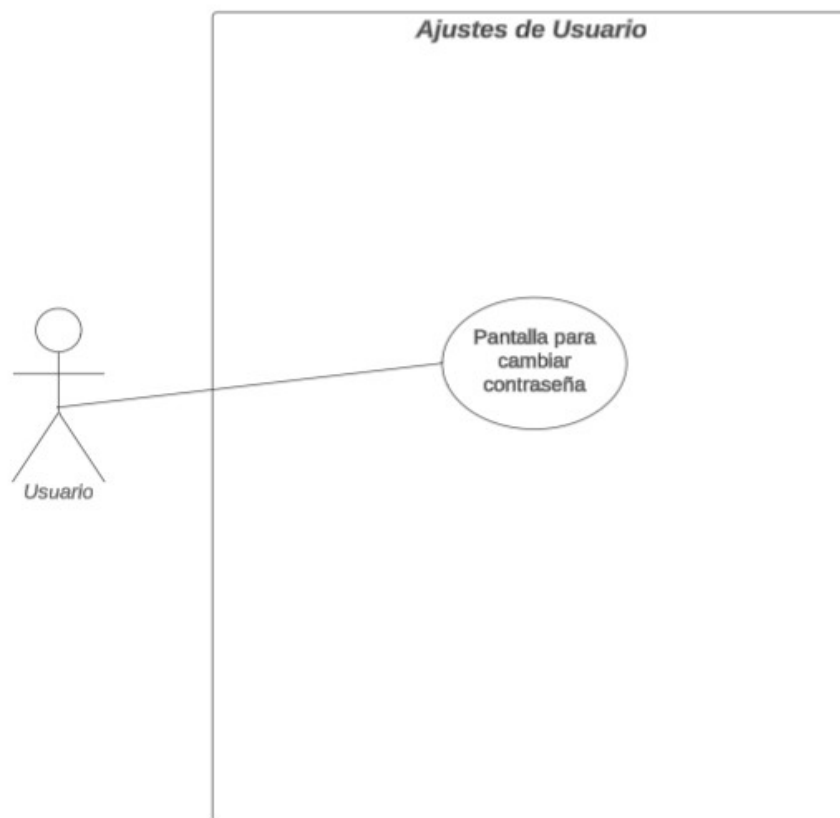
Nombre	CU10
Actor	Usuario
Descripción	El usuario cambia su foto de perfil
Evento desencadenado	La foto de perfil del usuario se cambia
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar botón 2. Se cambian los datos del usuario tanto en la UI como en la base de datos
Precondiciones	Estar en la pantalla de Ajustes de Usuario
Postcondiciones	Se abre un menu y el usuario cambia su foto de perfil
Vista	Ajustes de Usuario



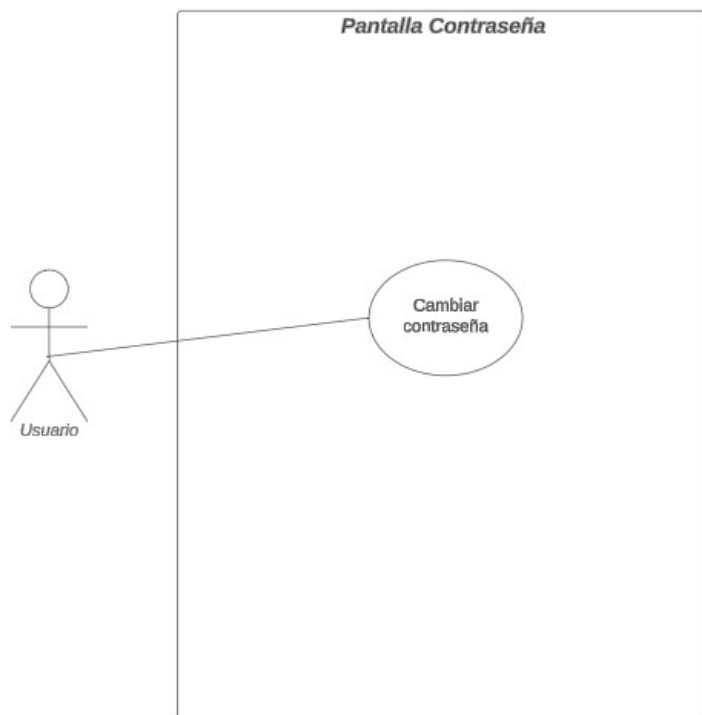
Nombre	CU10
Actor	Usuario
Descripción	El usuario cambia su nombre de usuario
Evento desencadenado	El nombre de usuario se cambia
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar botón 2. Se cambian los datos del usuario tanto en la UI como en la base de datos
Precondiciones	Estar en la pantalla de Ajustes de Usuario
Postcondiciones	Se abre un menu y el usuario cambia su nombre
Vista	Ajustes de Usuario



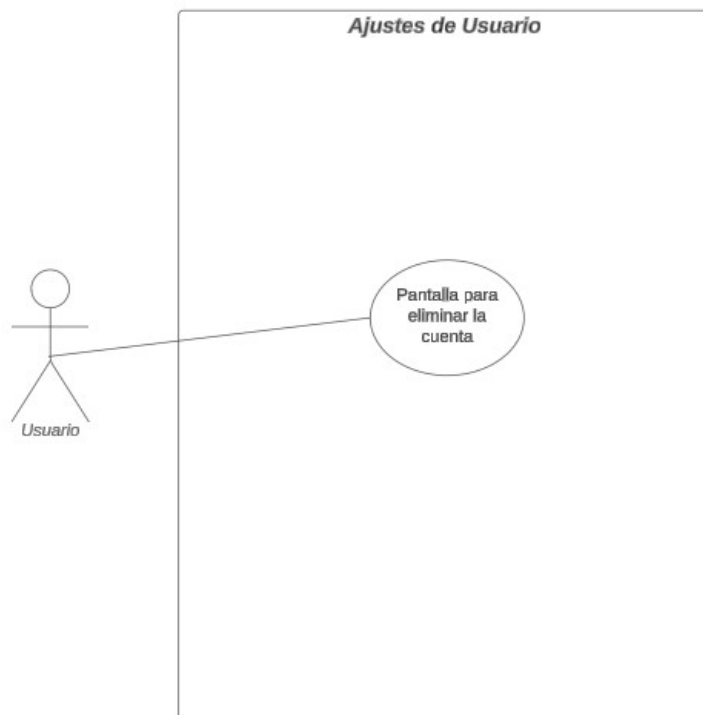
Nombre	CU11
Actor	Usuario
Descripción	El usuario cambia su email
Evento desencadenado	El email del usuario se cambia
Pasos realizados	<ol style="list-style-type: none">1. Pulsar botón2. Se cambian los datos del usuario tanto en la UI como en la base de datos
Precondiciones	Estar en la pantalla de Ajustes de Usuario
Postcondiciones	Se abre un menu y el usuario cambia su nombre
Vista	Ajustes de Usuario



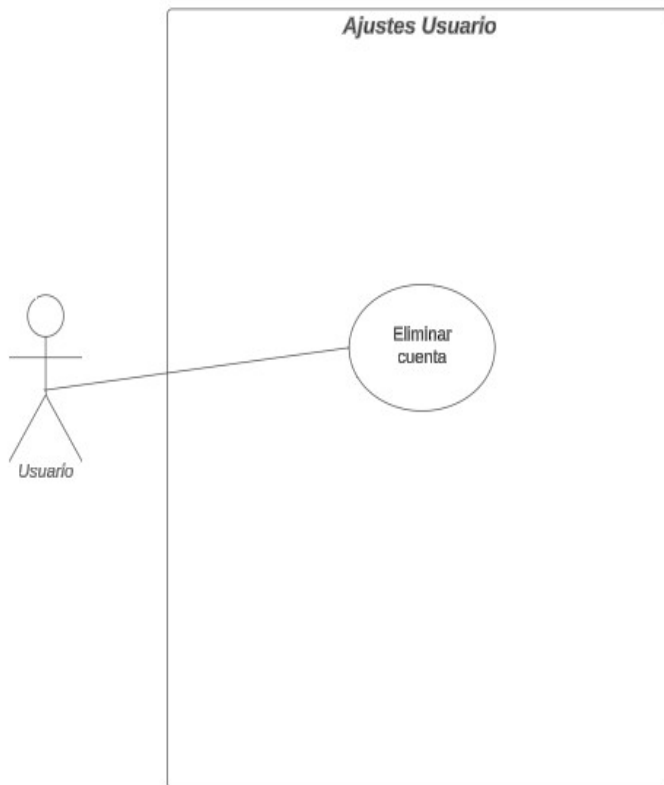
Nombre	CU12
Actor	Usuario
Descripción	El usuario se cambia de pantalla a la de Cambiar contraseña
Evento desencadenado	La pantalla se cambia a la de Cambiar contraseña
Pasos realizados	<ol style="list-style-type: none">1. Pulsar el botón2. Se cambia de pantalla de Cambiar contraseña
Precondiciones	Abrir la pantalla de Ajustes de Usuario
Postcondiciones	Cambiar de pantalla de Cambiar Contraseña
Vista	Ajustes de Usuario



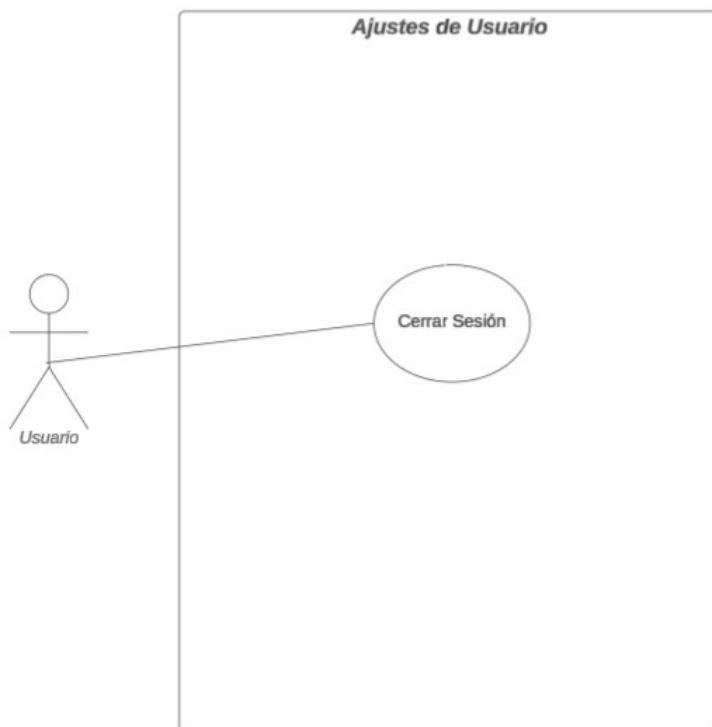
Nombre	CU13
Actor	Usuario
Descripción	El usuario cambia su contraseña
Evento desencadenado	La contraseña del usuario se cambia
Pasos realizados	<ol style="list-style-type: none">1. Pulsar botón2. Se cambian los datos del usuario tanto en la UI como en la base de datos
Precondiciones	Estar en la pantalla de Cambiar Contraseña
Postcondiciones	Se abre un menu y el usuario cambia su nombre
Vista	Cambiar Contraseña



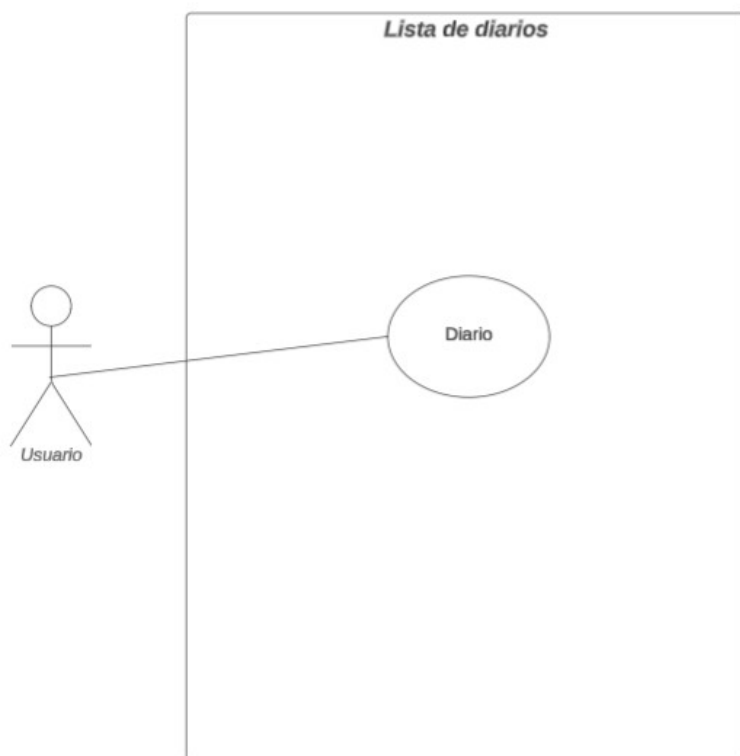
Nombre	CU14
Actor	Usuario
Descripción	El usuario se cambia de pantalla a la de Eliminar contraseña
Evento desencadenado	La pantalla se cambia a la de Eliminar contraseña
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar el botón 2. Se cambia de pantalla de Eliminar contraseña
Precondiciones	Abrir la pantalla de Ajustes de Usuario
Postcondiciones	Cambiar de pantalla de Cambiar Contraseña
Vista	Ajustes de Usuario



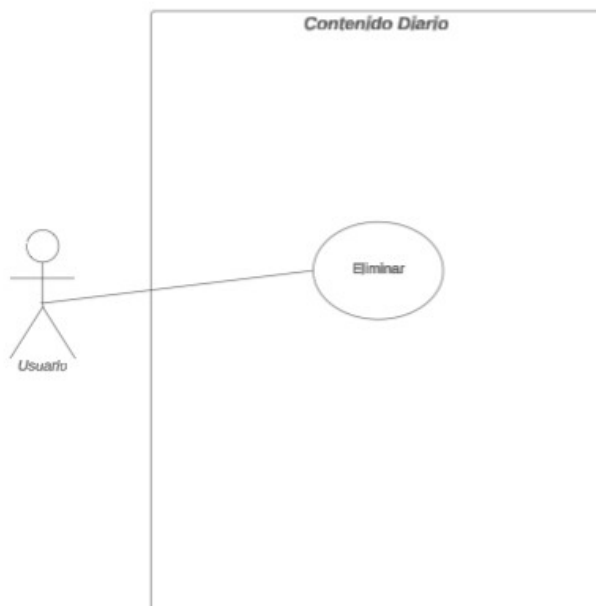
Nombre	CU15
Actor	Usuario
Descripción	El usuario elimina la cuenta
Evento desencadenado	Se despliega el menu para eliminar la cuenta
Pasos realizados	<ol style="list-style-type: none">1. Pulsar el botón2. Se eliminan los datos del usuario como en la base de datos y se cierra sesión
Precondiciones	Pulsar el botón
Postcondiciones	Se elimina la cuenta
Vista	Ajustes de Usuario



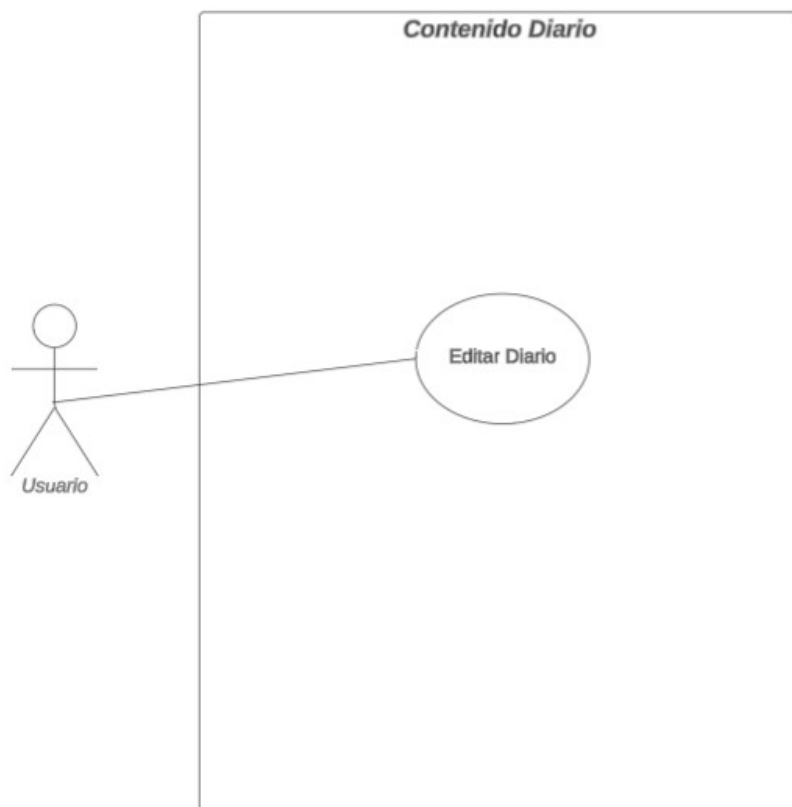
Nombre	CU16
Actor	Usuario
Descripción	El usuario cierra sesión
Evento desencadenado	Se despliega el menu para eliminar la cuenta
Pasos realizados	1. Pulsar el botón 2. Se elimina el token y se cierra sesión
Precondiciones	Pulsar el botón
Postcondiciones	Se cierra sesión de la cuenta
Vista	Ajustes de Usuario



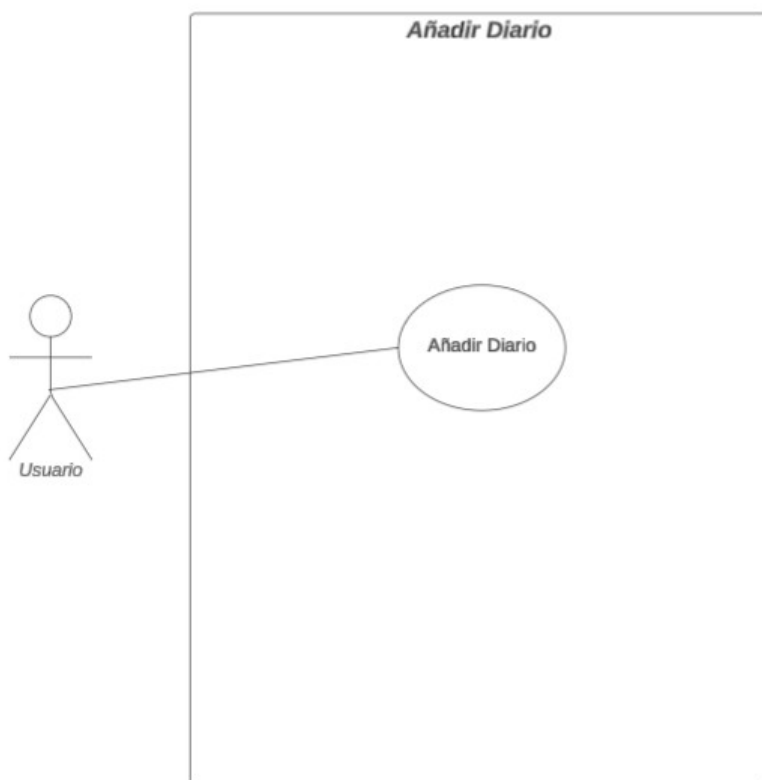
Nombre	CU17
Actor	Usuario
Descripción	El usuario se cambia de pantalla a la de Detalles Diarios
Evento desencadenado	La pantalla se cambia a la de Detalles Diarios
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar la tarea 2. Se cambia de pantalla de Detalles Tareas
Precondiciones	Abrir la pantalla de Lista de Diarios
Postcondiciones	Cambiar de pantalla de Detalles Diarios
Vista	Lista de Diarios



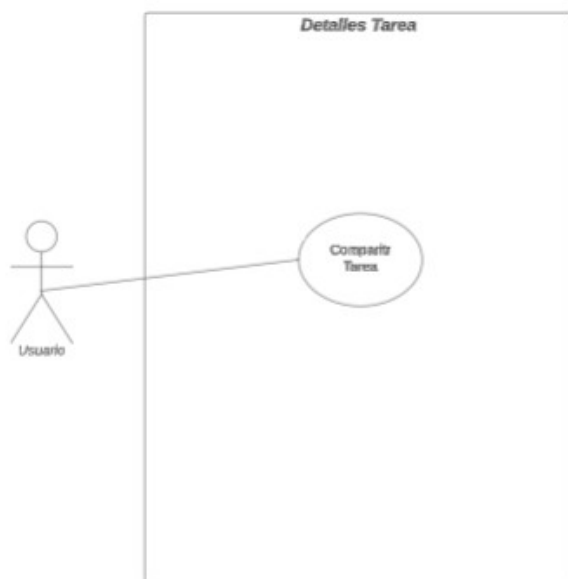
Nombre	CU18
Actor	Usuario
Descripción	El usuario elimina el diario en la pantalla de Detalles Diarios
Evento desencadenado	La pantalla se cambia a la de Lista Diarios y se elimina la tarea tanto en la UI como en la base de datos
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar el diario 2. Se elimina el diario 3. Se cambia de pantalla de Lista Diarios
Precondiciones	Abrir el Diario
Postcondiciones	Cambiar de pantalla de Lista de Diarios
Vista	Detalles Diarios



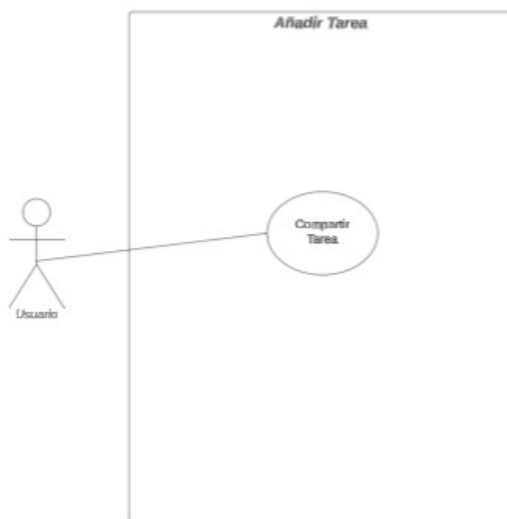
Nombre	CU19
Actor	Usuario
Descripción	El usuario edita el diario en la pantalla de Detalles Diarios
Evento desencadenado	La pantalla se cambia a la de Lista Diarios y se edita la tarea tanto en la UI como en la base de datos
Pasos realizados	<ol style="list-style-type: none">1. Pulsar el diario2. Se edita el diario3. Se cambia de pantalla de Lista Diarios
Precondiciones	Abrir el Diario
Postcondiciones	Cambiar de pantalla de Lista de Diarios
Vista	Detalles Diarios



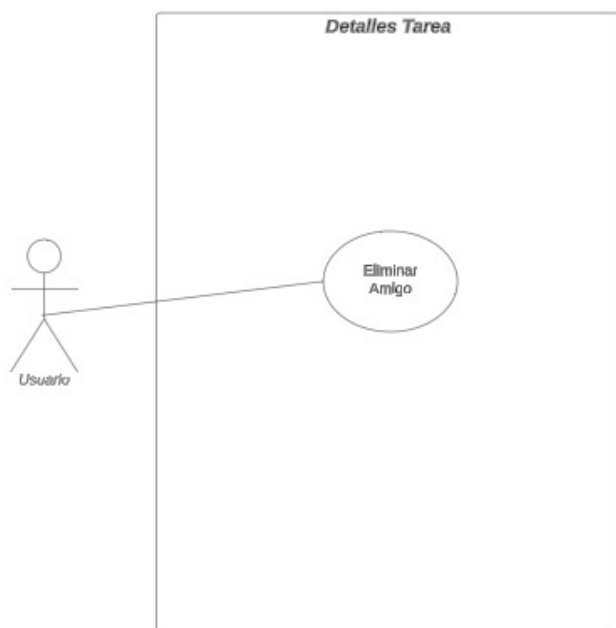
Nombre	CU20
Actor	Usuario
Descripción	El usuario se cambia la pantalla a la de Añadir Diarios
Evento desencadenado	La pantalla se cambia a la de Añadir Diarios
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar el botón 2. Se cambia de pantalla de Añadir Diarios
Precondiciones	Estar en la pantalla de Lista de Diarios
Postcondiciones	Cambiar de pantalla de Añadir Diarios
Vista	Añadir de Diarios



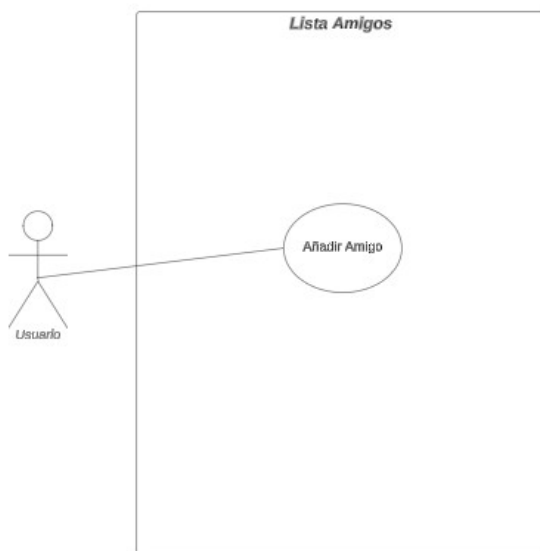
Nombre	CU21
Actor	Usuario
Descripción	El usuario comparte la tarea en la pantalla de Detalles Tareas
Evento desencadenado	La pantalla se cambia a la de Lista Tareas y se comparte la tarea. Esto se mostrará tanto en la UI como en la base de datos
Pasos realizados	<ol style="list-style-type: none">1. Pulsar el botón2. Se comparte la tarea3. Se cambia de pantalla de Lista Tareas
Precondiciones	Abrir la tarea
Postcondiciones	Cambiar de pantalla de Lista de Tareas
Vista	Detalles Tareas



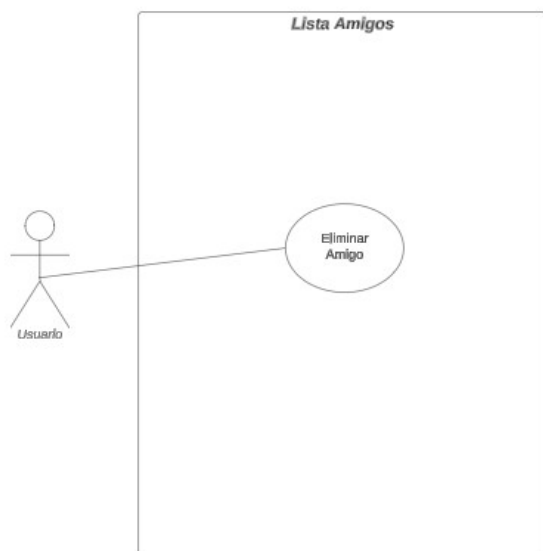
Nombre	CU22
Actor	Usuario
Descripción	El usuario comparte la tarea en la pantalla de Añadir Tareas
Evento desencadenado	La pantalla se cambia a la de Lista Tareas y se comparte la tarea. Esto se mostrará tanto en la UI como en la base de datos
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar el botón 2. Se comparte la tarea 3. Se cambia de pantalla de Lista Tareas
Precondiciones	Abrir la tarea
Postcondiciones	Cambiar de pantalla de Lista de Tareas
Vista	Detalles Tareas



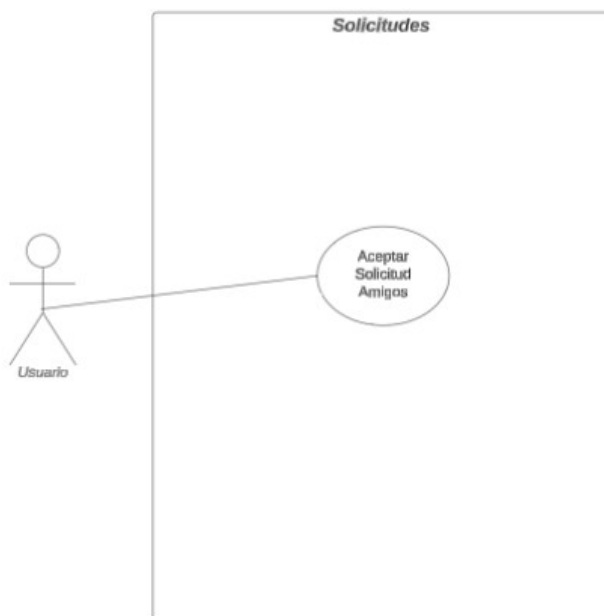
Nombre	CU23
Actor	Usuario
Descripción	El usuario elimina al amigo en la tarea en la pantalla de Detalles Tareas
Evento desencadenado	La pantalla se cambia a la de Lista Tareas y se elimina al amigo en la tarea. Esto se mostrará tanto en la UI como en la base de datos
Pasos realizados	<ol style="list-style-type: none">1. Pulsar el botón2. Se elimina la tarea3. Se cambia de pantalla de Lista Tareas
Precondiciones	Abrir la tarea
Postcondiciones	Cambiar de pantalla de Lista de Tareas
Vista	Detalles Tareas



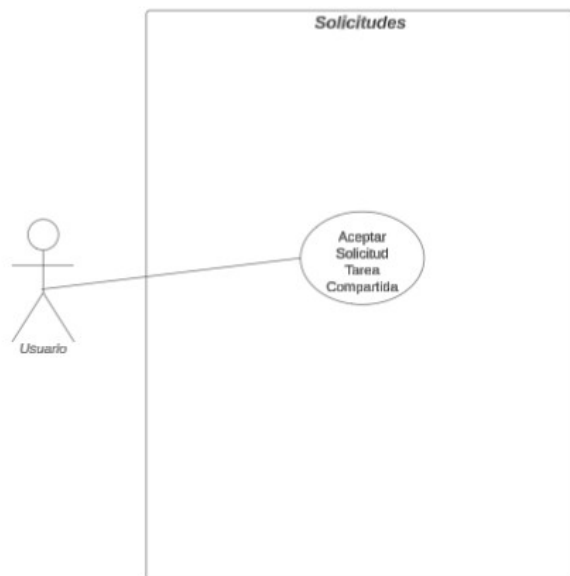
Nombre	CU24
Actor	Usuario
Descripción	El usuario añade una solicitud en la pantalla de Lista Amigos
Evento desencadenado	Se mandará una solicitud de amistad al usuario que se solicitará
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar el botón 2. Se añade la solicitud
Precondiciones	Estar en la pantalla de Lista Amigos
Postcondiciones	Se enviará una solicitud a un usuario
Vista	Lista Amigos



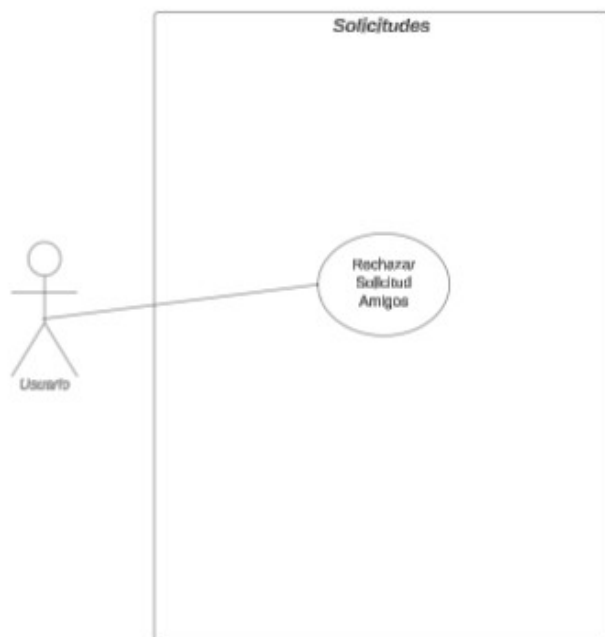
Nombre	CU25
Actor	Usuario
Descripción	El usuario elimina un amigo en la pantalla de Lista Amigos
Evento desencadenado	Se eliminará una amistad al usuario tanto en la UI como en la base de datos
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar el botón 2. Se eliminará la amistad
Precondiciones	Estar en la pantalla de Lista Amigos
Postcondiciones	Se enviará una solicitud a un usuario
Vista	Lista Amigos



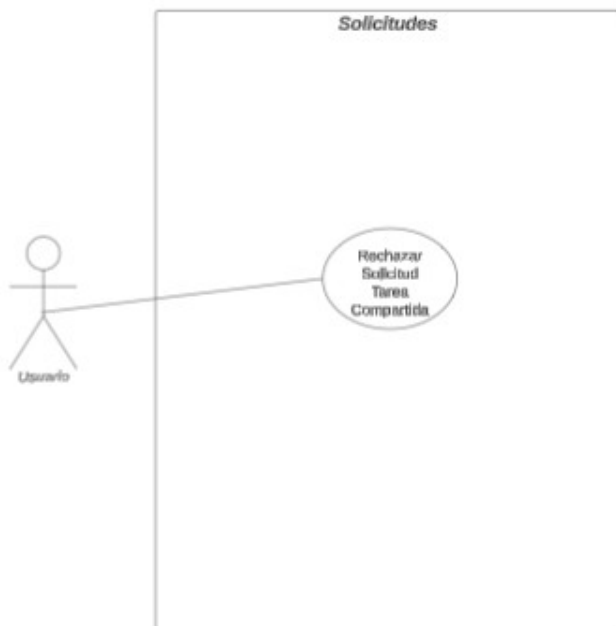
Nombre	CU26
Actor	Usuario
Descripción	El usuario aceptar a un amigo en la pantalla de Solicitudes
Evento desencadenado	Se añadirá una amistad al usuario tanto en la UI como en la base de datos
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar el botón 2. Se añadirá la amistad
Precondiciones	Estar en la pantalla de Solicitudes
Postcondiciones	Se aceptará una solicitud de un usuario
Vista	Solicitudes



Nombre	CU27
Actor	Usuario
Descripción	El usuario acepta la una solicitud de compartir tarea en la pantalla de Solicitudes
Evento desencadenado	Se aceptará una solicitud de compartir tarea al usuario tanto en la UI como en la base de datos
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar el botón 2. Se añadirá la tarea
Precondiciones	Estar en la pantalla de Solicitudes
Postcondiciones	Se aceptará una solicitud de compartir tarea
Vista	Solicitudes



Nombre	CU28
Actor	Usuario
Descripción	El usuario elimina a una solicitud en la pantalla de Solicitudes
Evento desencadenado	Se eliminará una solicitud de un usuario tanto en la UI como en la base de datos
Pasos realizados	1. Pulsar el botón 2. Se añadirá la amistad
Precondiciones	Estar en la pantalla de Solicitudes
Postcondiciones	Se eliminará una solicitud de un usuario
Vista	Solicitudes



Nombre	CU29
Actor	Usuario
Descripción	El usuario elimina una solicitud de compartir tarea en la pantalla de Solicitudes
Evento desencadenado	Se aceptará una solicitud de compartir tarea al usuario tanto en la UI como en la base de datos
Pasos realizados	<ol style="list-style-type: none"> 1. Pulsar el botón 2. Se eliminará la solicitud
Precondiciones	Estar en la pantalla de Solicitudes
Postcondiciones	Se eliminará una solicitud de compartir tarea
Vista	Solicitudes

4. OBJETIVOS

El objetivo general es crear una aplicación de gestión de tareas y objetivos intuitivos y eficientes para mejorar la productividad y la satisfacción del usuario al cumplir sus compromisos diarios y alcanzar sus objetivos a largo plazo.

Objetivos específicos:

1. Analizar las principales dificultades que enfrentan los usuarios en la gestión de sus tareas diarias y objetivos a largo plazo.
2. Definir los parámetros y funcionalidades necesarias para una aplicación que gestione tareas y objetivos de manera intuitiva y eficiente.
3. Optimizar la interfaz de usuario para asegurar una experiencia fluida y agradable durante la gestión de tareas y objetivos.

5. DESARROLLO

5.1. METODOLOGÍA

La metodología empleada fue la del método cascada, en la que se estableció una serie de tareas antes de iniciar el proceso de desarrollo de la aplicación. Durante este proceso, era necesario completar cada tarea antes de avanzar a la siguiente. También se estableció una temporalización, antes de realizar el proyecto la cual es la siguiente:

TEMPORALIZACIÓN SEMANAL DEL PROYECTO

Fecha: 14 de marzo – 5 de junio

Semana 14 - 20 marzo

- Inicio del backend.
- Desarrollo del modelo, rutas, validaciones y lógica inicial de TAREAS.
- Conexión con la base de datos y configuración de entorno.

Semana 21 - 27 marzo

- Continuación de TAREAS: edición, eliminación, validaciones más complejas.
- Implementación de filtros por fecha, estado y búsqueda por nombre.

Semana 28 marzo - 3 abril

- Desarrollo de USUARIOS: creación y autenticación con JWT.
- Creación de DIARIOS: modelo, controladores, y rutas principales.

Semana 4 - 10 abril

- Continuación de USUARIOS: validaciones, edición de perfil (no se va a hacer), y gestión de datos.
- Continuación de DIARIOS: edición y eliminación, validaciones y relaciones con usuario.

Semana 11 - 17 abril

- Módulo de AMIGOS: modelo, controladores y rutas para solicitudes.
- Funcionalidad de enviar, aceptar, rechazar y eliminar solicitudes de amistad.

Semana 25 abril

- Desarrollo de TAREAS COMPARTIDAS: asignación de tareas entre usuarios, validaciones y control de acceso.
- Conexión entre módulo de amigos y tareas compartidas.

Semana 26 abril - 2 mayo

- Pantallas de inicio de sesión y registro con validaciones.
- Conexión al backend para autenticación y persistencia de sesión.

Semana 3 - 9 mayo

- UI de TAREAS: listado principal, añadir tarea, mostrar estado, filtros.
- Diseño adaptativo y conexión con API de tareas.
- Detalle y edición de TAREAS.
- Eliminación y actualización dinámica de la lista.

Semana 10 - 16 mayo

- Detalle y edición de TAREAS.
- Eliminación y actualización dinámica de la lista.
- Interfaz de DIARIOS: crear, editar y mostrar entradas vinculadas al usuario.

Semana 17 - 23 mayo

- Interfaz de DIARIOS: crear, editar y mostrar entradas vinculadas al usuario.

Semana 24 - 30 mayo

- Módulo de AMIGOS: visualización, solicitudes entrantes/salientes y acciones.
- Conexión completa con la lógica de backend para gestión de amistades.

Semana 31 mayo - 5 junio

- Ajustes del perfil de usuario: modificación de datos, cierre de sesión.
- Implementación de vista de calendario integrada con las tareas.

LISTADO DE TAREAS

	ID de la Tarea	Nombre de la tarea	Tiempo (min)
	R4F1T1	Crear modelo de Tarea	180
	R4F1T2	Crear conexión con la Base de Datos de la tabla Tareas	480
	R4F1T3	Crear controlador de Tarea	360
	R4F1T4	Crear validaciones de Tarea	90
	R4F1T5	Crear las rutas para la conexión (GET, POST, PUT, DELETE) de Tarea	360
	R4F1T6	Implementar el modelo, el controlador, las validaciones y las rutas de Tarea en el servidor	180
	R5F1T1	Crear modelo de Usuario	240
	R5F1T2	Crear conexión con la Base de Datos de la tabla Usuarios	240
	R5F1T3	Crear controlador de Usuario	240
	R5F1T4	Crear validaciones de Usuario	240
	R5F1T5	Crear las rutas para la conexión (GET, POST, PUT, DELETE) de Usuario	120
	R5F1T6	Implementar el modelo, el controlador, las validaciones y las rutas de Usuario en el servidor	240
	R3F1T1	Crear modelo de Diario	60
	R3F1T2	Crear conexión con la Base de Datos de la tabla Diario	60
	R3F1T3	Crear controlador de Diario	60
	R3F1T4	Crear validaciones de Diario	60
	R3F1T5	Crear las rutas para la conexión (GET, POST, PUT, DELETE) de Diario	30
	R3F1T6	Implementar el modelo, el controlador, las validaciones y las rutas de Diario en el servidor	60
	R1F1T1	Crear UI de Inicio de Sesión	120
	R1F1T2	Crear funciones para iniciar sesión	120
	R1F2T1	Crear botón para cambiar de pantalla de Registro	120
	R2F1T1	Crear UI de Registro	120
	R2F1T2	Crear funciones para registrarse	10
	R2F1T3	Crear validaciones de contraseña	10
	R2F2T1	Crear botón para cambiar de pantalla	60
	R4F2T1	Cargar tareas desde la base de datos	60
	R4F2T2	Hacer que las tareas muestren el título, la fecha y la categoría	60
	R4F2T3	Devolver mensaje si no cargan las tareas	10

R4F2T4	Mostrar que el usuario no tiene tareas	10
R4F4T1	Mostrar los detalles de la tarea desde la lista de tareas	10
R4F4T2	Hacer botón de editar en los detalles	120
R4F4T3	Hacer botón de eliminar en los detalles	120
R4F5T1	Hacer validaciones antes de editar la tarea	120
R4F5T2	Hacer un PUT en el servidor de la tarea	30
R4F5T3	Hacer que se cambie la vista de la lista de tareas al editar	30
R4F6T1	Hacer un DELETE en el servidor de la tarea	60
R4F6T2	Hacer que se cambie la vista de la lista de tareas al eliminar	10
R4F7T1	Hacer botón de Añadir Tarea	10
R4F7T2	Hacer validaciones antes de añadir la tarea	60
R4F7T3	Añadir la tarea en la lista de tareas:	60
R4F7T4	Hacer un POST de la tarea al servidor	60
<u>R3F2T1</u>	<u>Cargar diarios desde la base de datos</u>	60
<u>R3F2T2</u>	<u>Mostrar la fecha del diario y título</u>	60
<u>R3F2T3</u>	<u>Devolver mensaje si no cargan los diarios</u>	10
<u>R3F2T4</u>	<u>Mostrar mensaje de que el usuario no tiene diarios</u>	10
<u>R3F3T1</u>	<u>Crear botón de editar diario</u>	60
<u>R3F3T2</u>	<u>Crear botón de eliminar diario</u>	60
R3F4T1	Hacer validaciones antes de editar el diario	10
R3F4T2	Permitir que se edite al mostrar el contenido	10
R3F4T3	Hacer un PUT del diario en el servidor	30
R3F5T1	Permitir que el botón de eliminar borre al mostrar el contenido del diario	30
R3F5T2	Hacer un DELETE del diario en el servidor	30
R4F6T1	Hacer validaciones antes de añadir el diario	10
R4F6T2	Añadir el diario en la lista de diarios	10
R4F6T3	Hacer un POST del diario en el servidor	10
R5F2T1	Crear la función para cambiar el nombre	120
R5F2T2	Crear la interfaz y el proceso para cambiar el nombre	120
R5F3T1	Crear la función para cambiar la contraseña	120

	R5F3T2	Crear la interfaz y el proceso para cambiar la contraseña	120
	R5F4T1	Crear un formulario para cambiar o escoger una imagen desde los archivos del ordenador	0
	R5F4T2	Crear la funcionalidad para subir la imagen en el Servidor	0
	R5F4T3	Establecer la imagen que quiere subir el usuario como su foto de perfil	0
	R5F5T1	Crear las funciones para la persistencia de datos	360
	R5F6T1	Crear UI para cambiar el email	120
	R5F6T2	Crear funciones para cambiar el email.	120
	R5F7T1	Crear botón 'CERRAR SESIÓN'	10
	R5F7T2	Crear función para cerrar sesión	30
	R5F8T1	Crear botón y UI para eliminar cuenta	0
	R5F8T2	Crear función para eliminar la cuenta	50
	R6F1T1	Diseño y desarrollo de interfaz adaptable.	360
	R8F1T1	Crear modelo Amigo	0
	R8F1T2	Crear conexión con la Base de Datos de la tabla Amigos	0
	R8F1T3	Crear controlador de Amigo	0
	R8F1T4	Crear validaciones de Amigo	0
	R8F1T5	rear las rutas para la conexión (GET, POST, PUT, DELETE) de Amigos	0
	R8F1T6	Implementar el modelo, el controlador, las validaciones y las rutas de Amigos en el servidor	0
	R4F3T1	Cargar tareas desde la base de datos al calendario	0
	R4F4T4	Mostrar los detalles de la tarea desde el calendario	0
	R4F5T4	Hacer que se cambie la vista del calendario al editar la tarea	0
	R4F6T3	Hacer que se cambie la vista del calendario al eliminar la tarea	0
	R4F7T4	Añadir tarea en el calendario	0
	R4F11T1	Crear apartado de tipos de tarea	0
	R4F8T1	Crear modelo de Tareas Compartidas	0
	R4F8T2	Crear conexión con la Base de Datos de la tabla Tareas Compartidas	0
	R4F8T3	Crear controlador de Tareas Compartidas	0
	R4F8T4	Crear validaciones de Tareas Compartidas	0
	R4F8T5	Crear las rutas para la conexión (GET, POST, PUT, DELETE) de Tareas Compartidas	0
	R4F8T6	Implementar el modelo, el controlador, las validaciones y las rutas de Tareas Compartidas en el servidor	0

	R4F9T4	Cambiar esa tarea a compartida en la base de datos	0
	R8F2T1	Crear UI de lista de amigos	0
	R8F2T2	Cargar amigos desde la base de datos	0
	R8F3T1	Crear UI para eliminar amigo	0
	R8F3T2	Conectarlo a base de datos	0
	R8F4T1	Crear UI para añadir amigo	0
	R8F4T2	Crear UI para añadir amigo	0
	R8F5T1	Crear Menu de filtrado de Solicitudes	0
	R8F5T2	Crear UI de solicitudes de Amigos	0
	R8F5T3	Crear UI de solicitudes de Tareas	0
	R4F9T1	Hacer botón de Compartir Tarea	0
	R4F10T1	Hacer botón de Elimnar Amigo Tarea	0
	R4F10T4	Eliminar ese amigo en la tarea en la base de datos	0
	R8F5T4	Crear UI de solicitudes de Todos	0
	R8F6T1	Crear botón para eliminar solicitud de Amigos	0
	R8F6T2	Eliminar solicitud de Amigos en base de datos	0
	R8F6T3	Crear botón para eliminar solicitud de Tareas	0
	R8F6T4	Comprobar si la solicitud se ha eliminado en la base de datos	0
	Tiempo Total		6620

Diagrama de Gantt

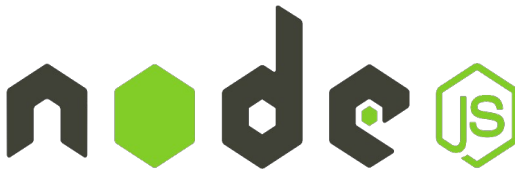
Nombre de la tarea	Fecha de inicio	Fecha de finalización	Asignado	Estado
	14.03.2025	15.03.2025	Neil	Cerrado
Crear modelo de Tarea	15.03.2025	18.03.2025	Neil	Cerrado
Crear conexión con la Base de Datos de la tabla Tareas	18.03.2025	19.03.2025	Neil	Cerrado
Crear controlador de Tarea	19.03.2025	22.03.2025	Neil	Cerrado
Crear validaciones de Tarea	22.03.2025	24.03.2025	Neil	Cerrado
Crear las rutas para la conexión (GET, POST, PUT, DELETE) de Tarea	24.03.2025	26.03.2025	Cerrado	
Implementar el modelo, el controlador, las validaciones y las rutas de Tarea en el servidor	26.03.2025	26.03.2025	Cerrado	
Crear modelo de Usuario	26.03.2025	26.03.2025	Cerrado	
Crear conexión con la Base de Datos de la tabla Usuarios	26.03.2025	26.03.2025	Cerrado	
Crear controlador de Usuario	26.03.2025	29.03.2025	Cerrado	
Crear validaciones de Usuario	29.03.2025	31.03.2025	Cerrado	
Crear las rutas para la conexión (GET, POST, PUT, DELETE) de Usuario	31.03.2025	03.04.2025	Cerrado	
Implementar el modelo, el controlador, las validaciones y las rutas de Usuario en el servidor	03.04.2025	04.04.2025	Cerrado	
Crear modelo de Diario	05.04.2025	05.04.2025	Cerrado	
Crear conexión con la Base de Datos de la tabla Diario	05.04.2025	07.04.2025	Cerrado	
Crear controlador de Diario	07.04.2025	07.04.2025	Cerrado	
Crear validaciones de Diario	07.04.2025	08.04.2025	Cerrado	
Crear las rutas para la conexión (GET, POST, PUT, DELETE) de Diario	08.04.2025	08.04.2025	Cerrado	
Implementar el modelo, el controlador, las validaciones y las rutas de Diario en el servidor	09.04.2025	09.04.2025	Cerrado	
Crear UI de Inicio de Sesión	09.04.2025	13.04.2025	Cerrado	
Crear funciones para iniciar sesión	13.04.2025	17.04.2025	Cerrado	
Crear botón para cambiar de pantalla de Registro	17.04.2025	19.04.2025	Cerrado	
Crear UI de Registro	19.04.2025	20.04.2025	Cerrado	
Crear funciones para registrarse	21.04.2025	25.04.2025	Cerrado	
Crear validaciones de contraseña	25.04.2025	26.04.2025	Cerrado	
Crear botón para cambiar de pantalla	27.04.2025	16.05.2025	Cerrado	
Cargar tareas desde la base de datos	27.04.2025	16.05.2025	Cerrado	
Hacer que las tareas muestren el título, la fecha y la categoría	27.04.2025	16.05.2025	Cerrado	
Devolver mensaje si no cargan las tareas	27.04.2025	16.05.2025	Cerrado	
Mostrar que el usuario no tiene tareas	27.04.2025	16.05.2025	Cerrado	
Mostrar los detalles de la tarea desde la lista de tareas	27.04.2025	16.05.2025	Cerrado	
Hacer botón de editar en los detalles	17.05.2025	18.05.2025	Cerrado	
Hacer botón de eliminar en los detalles	18.05.2025	19.05.2025	Cerrado	
Hacer validaciones antes de editar la tarea	19.05.2025	20.05.2025	Cerrado	
Hacer un PUT en el servidor de la tarea	21.05.2025	23.05.2025	Cerrado	
Hacer que se cambie la vista de la lista de tareas al editar	21.05.2025	23.05.2025	Cerrado	
Hacer un DELETE en el servidor de la tarea	21.05.2025	23.05.2025	Cerrado	
Hacer que se cambie la vista de la lista de tareas al eliminar	21.05.2025	23.05.2025	Cerrado	
Hacer botón de Añadir Tarea	21.05.2025	23.05.2025	Cerrado	
Hacer validaciones antes de añadir la tarea	21.05.2025	23.05.2025	Cerrado	
Añadir la tarea en la lista de tareas:	21.05.2025	23.05.2025	Cerrado	
Hacer un POST de la tarea al servidor	21.05.2025	23.05.2025	Cerrado	
Cargar diarios desde la base de datos	24.05.2025	25.05.2025	Cerrado	
Mostrar la fecha del diario y título	24.05.2025	25.05.2025	Cerrado	
Devolver mensaje si no cargan los diarios	24.05.2025	25.05.2025	Cerrado	
Mostrar mensaje de que el usuario no tiene diarios	24.05.2025	25.05.2025	Cerrado	
Crear botón de editar diario	24.05.2025	25.05.2025	Cerrado	
Crear botón de eliminar diario	24.05.2025	25.05.2025	Cerrado	
Hacer validaciones antes de editar el diario	24.05.2025	25.05.2025	Cerrado	
Permitir que se edite al mostrar el contenido	24.05.2025	25.05.2025	Cerrado	
Hacer un PUT del diario en el servidor	24.05.2025	25.05.2025	Cerrado	
Permitir que el botón de eliminar borre al mostrar el contenido del diario	24.05.2025	25.05.2025	Cerrado	
Hacer un DELETE del diario en el servidor	24.05.2025	25.05.2025	Cerrado	
Hacer validaciones antes de añadir el diario	24.05.2025	25.05.2025	Cerrado	
Añadir el diario en la lista de diarios	24.05.2025	25.05.2025	Cerrado	
Hacer un POST del diario en el servidor	24.05.2025	25.05.2025	Cerrado	
Crear la función para cambiar el nombre	06.06.2025	08.06.2025	Cerrado	
Crear la interfaz y el proceso para cambiar el nombre	07.06.2025	08.06.2025	Cerrado	
Crear la función para cambiar la contraseña	06.06.2025	08.06.2025	Cerrado	
Crear la interfaz y el proceso para cambiar la contraseña	07.06.2025	08.06.2025	Cerrado	
Crear un formulario para cambiar o escoger una imagen desde los archivos del ordenador			En progreso	
Crear la funcionalidad para subir la imagen en el servidor			En progreso	
Establecer la imagen que quiere subir el usuario como su foto de perfil			En progreso	
Crear las funciones para la persistencia de datos			En progreso	
Crear UI para cambiar el email	07.06.2025	08.06.2025	Cerrado	
Crear funciones para cambiar el email.	06.06.2025	08.06.2025	Cerrado	
Crear botón 'CERRAR SESIÓN'	25.05.2025	26.05.2025	Cerrado	
Crear función para cerrar sesión	25.05.2025	29.05.2025	Cerrado	
Crear botón y UI para eliminar cuenta	26.05.2025	29.05.2025	Cerrado	
Crear función para eliminar la cuenta	26.05.2025	29.05.2025	Cerrado	
Diseño y desarrollo de interfaz adaptable.	30.05.2025	05.06.2025	Cerrado	
Crear modelo Amigo			En progreso	
Crear conexión con la Base de Datos de la tabla Amigos			En progreso	
Crear controlador de Amigo			En progreso	
Crear validaciones de Amigo			En progreso	
Crear las rutas para la conexión (GET, POST, PUT, DELETE) de Amigos			En progreso	
Implementar el modelo, el controlador, las validaciones y las rutas de Amigos en el servidor			En progreso	
Cargar tareas desde la base de datos al calendario			En progreso	
Mostrar los detalles de la tarea desde el calendario			En progreso	
Hacer que se cambie la vista del calendario al editar la tarea			En progreso	
Hacer que se cambie la vista del calendario al eliminar la tarea			En progreso	
Añadir tarea en el calendario			En progreso	
Crear apartado de tipos de tarea			En progreso	
Crear modelo de Tareas Compartidas			En progreso	
Crear conexión con la Base de Datos de la tabla Tareas Compartidas			En progreso	
Crear controlador de Tareas Compartidas			En progreso	
Crear validaciones de Tareas Compartidas			En progreso	
Crear las rutas para la conexión (GET, POST, PUT, DELETE) de Tareas Compartidas			En progreso	
Implementar el modelo, el controlador, las validaciones y las rutas de Tareas Compartidas en el servidor			En progreso	
Cambiar esa tarea a compartida en la base de datos			En progreso	
Crear UI de lista de amigos			En progreso	
Cargar amigos desde la base de datos			En progreso	
Crear UI para eliminar amigo			En progreso	
Conectarlo a base de datos			En progreso	
Crear UI para añadir amigo			En progreso	
Crear UI para añadir amigo			En progreso	
Crear Menu de filtrado de Solicitudes			En progreso	
Crear UI de solicitudes de Amigos			En progreso	
Crear UI de solicitudes de Tareas			En progreso	
Hacer botón de Compartir Tarea			En progreso	
Hacer botón de Eliminar Amigo Tarea			En progreso	
Eliminar ese amigo en la tarea en la base de datos			En progreso	
Crear UI de solicitudes de Todos			En progreso	
Crear botón para eliminar solicitud de Amigos			En progreso	
Eliminar solicitud de Amigos en base de datos			En progreso	
Crear botón para eliminar solicitud de Tareas			En progreso	
Comprobar si la solicitud se ha eliminado en la base de datos			En progreso	

5.2. TECNOLOGÍAS

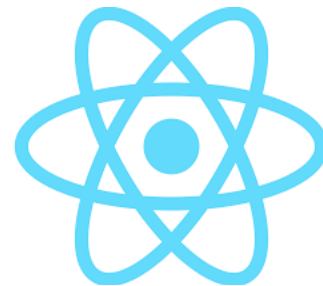
AWS: Se ha escogido por su facilidad de tener la aplicación en la nube. Con sus servicios como RDS para guardar la base de datos MySQL y EC2 para alojar el backend, facilitan el despliegue de la página web.



NodeJS: Se utiliza como tecnología de backend por su capacidad para gestionar la transferencia de datos hacia la base de datos. Se aprovechan herramientas como Express para la creación de la API-REST, Zod para realizar validaciones de datos y Socket.io para crear los WebSocket (Midudev, s. f.)

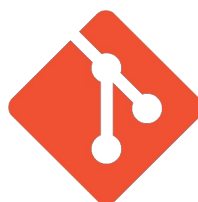


React: Una de las bibliotecas de Javascript más usadas en el desarrollo frontend, por la facilidad que ofrece a la hora de programar con sus bibliotecas como **React-Router** para las rutas, **Redux** para la gestión de estados globales o **Axios** para hacer peticiones fetch al backend de manera más legible. También ofrece la creación de componentes, que hace que el código sea más escalable.



TypeScript: Se prefiere sobre JavaScript en el frontend por su tipado estático, lo que brinda mayor seguridad y claridad en el código, especialmente para desarrolladores acostumbrados a lenguajes tipados como Java. Esto contribuye a reducir la probabilidad de errores y a mejorar la mantenibilidad del código.

Git y GitHub: tecnologías para el control de versiones, he utilizado dado que son las tecnologías más utilizadas en esta área.





5.3. PROTOTIPADO Y USABILIDAD

PROTOTIPOS

Para los prototipos se ha utilizado **Figma**.

Login:

The image shows a login form prototype on a dark blue background. The form is a light gray rectangle with rounded corners. At the top, it says "Inicio de Sesión" in bold black text. Below that, it says "Bienvenido" in bold black text. Then, it asks "¿Tienes cuenta de Google?" in a smaller font. There are two white input fields with blue borders, one for email and one for password. Below the input fields is a blue button with the text "Acceder". At the bottom, it says "¿Tienes cuenta? [Registrate](#)" in blue text.



Registro:

Registro


Bienvenido

Create una cuenta para continuar

Acceder

¿Tienes cuenta? [Inicia Sesión](#)

Lista de Tareas:

 TASK-CRAFT

🔍 Buscar

Tareas ▾

Tipo 1

Tipo 2


Tipo 3

Añadir +

📅 Calendario

📅 Diarios

👤 Amigos

 Usuario

Tareas

Añadir Tarea

Tipo 1

Tarea 1

● Tipo 1

20/10/2025

11:30

Tarea 2

● Tipo 1

23/12/2024

12:15

Tipo 2

Tarea 3

● Tipo 2

25/01/2025

23:00

Tarea 4

● Tipo 2

12/02/2025

20:00

Tipo 3

Tarea 5

● Tipo 3

25/01/2025

23:00

Tarea 6

● Tipo 3

12/02/2025

20:00

Tarea 7

● Tipo 3

25/01/2025

23:00

Tarea 8

● Tipo 3

12/02/2025

20:00

Página 48 de 88



Detalles Tareas:

Ver Amigos

Compartir Tarea

Titulo 0/25

dd/mm/yyyy

hh:mm

Fecha

Hora

Descripción 0/250

Categoría

Tipo 1 ▼

Estado

Estado 1 ▼

Eliminar

Editar



Añadir Tareas

Compartir Tarea

Titulo 0/25

dd/mm/yyyy

hh:mm

Fecha

Hora

Descripción 0/250

Categoría

Tipo 1

Estado

Estado 1

Añadir



Calendario (se utiliza componente de Javascript):

TASK-CRAFT

Buscar

Tareas ▾

Tipo 1

Tipo 2

Tipo 3

Añadir +

Calendario

Diarios

Amigos

Usuario



Lista de Diarios:

Buscar

Tareas ▾

Tipo 1

Tipo 2

Tipo 3

Añadir +

Calendario

Diarios

Amigos

Usuario

Diarios

Añadir Diario

Diario 1

5 de enero de 2025

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. ...

Diario 2

1 de enero de 2025

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. ...



Detalles de Diarios:

Titulo

Eliminar

Editar

Dia de mes de año

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. ...



Añadir Diario:

Titulo

Añadir

Dia de mes de año

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. ...



Lista de Amigos:

Buscar

Tareas ▾

Tipo 1

Tipo 2

Tipo 3

Añadir +

Calendario

Diarios

Amigos

Usuario

Amigos

Añadir Amigos

Lista de Amigos Solicitudes Solicitudes Enviadas

Nombre del Amigo

Nombre del Amigo



Solicitudes Todas:

Buscar

Tareas ▾

Tipo 1

Tipo 2

Tipo 3

Añadir +

Calendario

Diarios

Amigos

Usuario

Amigos

Lista de Amigos **Solicitudes** Solicitudes Enviadas

Todas

Amigos

Tareas

Amigos

Nombre del Amigo

✓

Nombre del Amigo

✓

Tareas

Amigo Quiere Compartir NombreDeLaTarea

✓

Página 56 de 88



Solicitudes de amigos:

Buscar

Tareas ▾

Tipo 1

Tipo 2

Tipo 3

Añadir +

Calendario

Diarios

Amigos

Usuario

Amigos

Lista de Amigos **Solicitudes** Solicitudes Enviadas

Todas **Amigos** Tareas

Nombre del Amigo

✓

Nombre del Amigo

✓



Solicitudes de Amigos:

Buscar

Tareas ▾

Tipo 1

Tipo 2

Tipo 3

Añadir +

Calendario

Diarios

Amigos

Usuario

Amigos

Lista de Amigos **Solicitudes** Solicitudes Enviadas

Todas

Amigos

Tareas

Amigo Quiere Compartir NombreDeLaTarea



Solicitudes Enviadas:

Buscar

Tareas ▾

Tipo 1

Tipo 2

Tipo 3

Añadir +

Calendario

Diarios

Amigos

Usuario

Amigos

Lista de Amigos Solicitudes **Solicitudes Enviadas**

Todas

Amigos

Tareas

Quieres compartir nombreTarea a nombreAmigo

Cancelar



Ajustes de Usuario:

TASK-CRAFT

Buscar

Tareas ▾

Tipo 1

Tipo 2

Tipo 3

Añadir +

Calendario

Diarios

Amigos

Usuario

Ajustes del Usuario

Detalles del Perfil

Subir foto de perfil

Nombre de usuario

Usuario

Cambiar

Email

usuario@gmail.com

Cambiar

Cambiar contraseña

Para cambiar tu contraseña, haz clic en el botón 'Cambiar contraseña'. Sigue las instrucciones que aparecerán para actualizarla de manera rápida y segura.

Cambiar Contraseña

Cerrar Sesión

Si deseas cerrar sesión, haz clic en el botón 'Cerrar sesión'. Esto te desconectará de forma segura y protegerá tu información, especialmente en dispositivos compartidos o públicos.

Cerrar Sesión

Eliminar Cuenta

Si deseas eliminar tu cuenta, haz clic en el botón 'Eliminar cuenta'. Ten en cuenta que esta acción es permanente y no podrá deshacerse.

Eliminar Cuenta

Parte de administración

Como administrador, puedes gestionar las cuentas de los usuarios, modificar configuraciones globales y supervisar la actividad del sistema. Asegúrate de realizar estas acciones con precaución para mantener la seguridad e integridad de la plataforma

Acceder



Parte de Admin:

Buscar

Tareas ▾

Tipo 1
Tipo 2
Tipo 3
Añadir +

Calendario

Diarios

Amigos

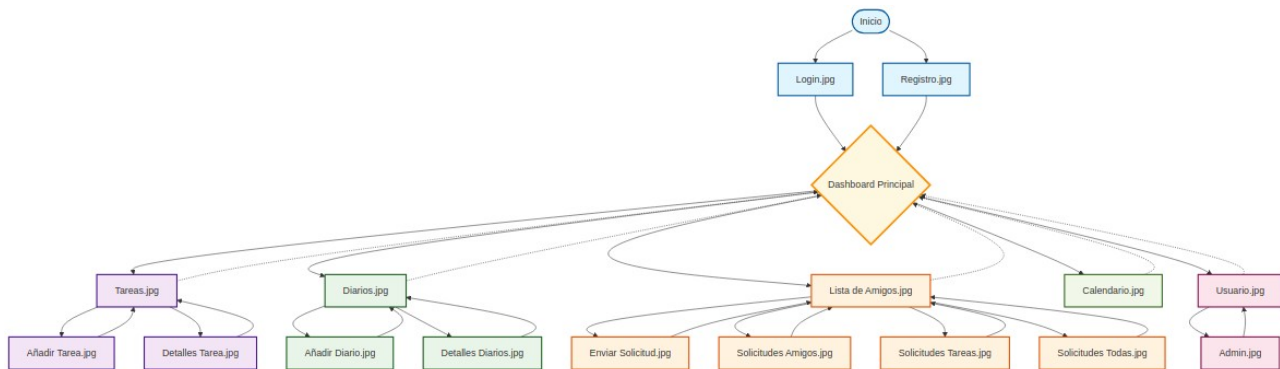
Usuario

Administración

Usuarios Tareas Diarios Tipos

Nombre	Email	Contraseña	Url-Foto-Perfil	Acciones
Usuario 1	usuario@gmail.com	48be4e6348...	https://example.com/images/...	

DIAGRAMA DE FLUJO:



USABILIDAD:

Diseño enfocado en la persona

La disposición del sistema presenta una navegación intuitiva en la que el menú lateral, siempre visible, permite un rápido acceso a las secciones más relevantes: diario, amigos, administración, configuraciones, etc.

Interfaz sencilla

La combinación de colores entre azul, blanco y negro hace que resalte el contenido y evita la sobrecarga visual.

Las pantallas están dispuestas con un espacio bien configurado entre cada uno de los elementos, permitiendo a los usuarios concentrarse a la hora de asumir sus tareas.

Flujo de tareas limpio

Procesos tales como el **registro**, el **login**, la creación de entradas del diario y la gestión de amigos están bien separados en formularios simples y accesibles.

Las acciones como **"Enviar"**, **"Aceptar"**, **"Rechazar"**, **"Eliminar"** están bien posicionadas, utilizando gráficos y colores (verde, rojo) que facilitan su comprensión.

Feedback inmediato y control del usuario

En el apartado de amigos se proporciona un feedback evidente y rápido al enviar una solicitud de amistad o al aceptar una solicitud de amistad.

En los ajustes de usuario y de administración se proporciona confirmación e independencia entre las opciones (por ejemplo, cambio de contraseña o eliminación de cuenta).

Accesibilidad y homogeneidad

La utilización homogénea de botones, campos de introducción y tipografía, ayuda al usuario en la adaptación rápida a su entorno.

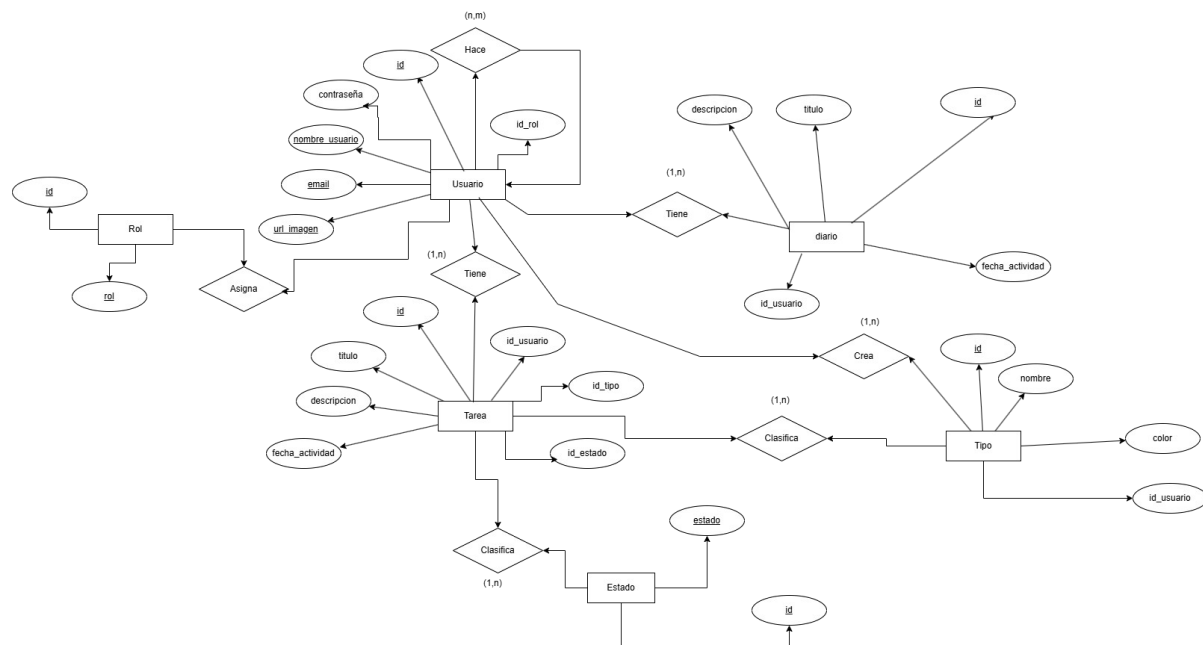
Optimización para los diferentes niveles de usuario

Los usuarios principiantes y avanzados pueden navegar sin una curva de aprendizaje pronunciada.

La interfaz busca la simplicidad y minimalismo, permitiendo una experiencia de usuario fluida y eficiente desde el primer uso.

5.4. MODELADO DE DATOS

Modelo Entidad Relación:



Entidades:

- Entidad usuario:
 - id: Clave primaria (Identificador único del usuario).
 - nombre: Nombre del usuario.
 - email: Dirección de correo electrónico del usuario. Atributo único.
 - url_imagen: URL que lleva a la imagen del usuario. Atributo único.
 - contraseña: Contraseña del usuario.
 - idRol: Clave foránea que hacer referencia al id de la tabla rol. Esto establece el rol del usuario (admin, user, etc.).
- Entidad rol:
 - id: Clave primaria (Identificador único del rol).
 - rol: Nombre del rol.

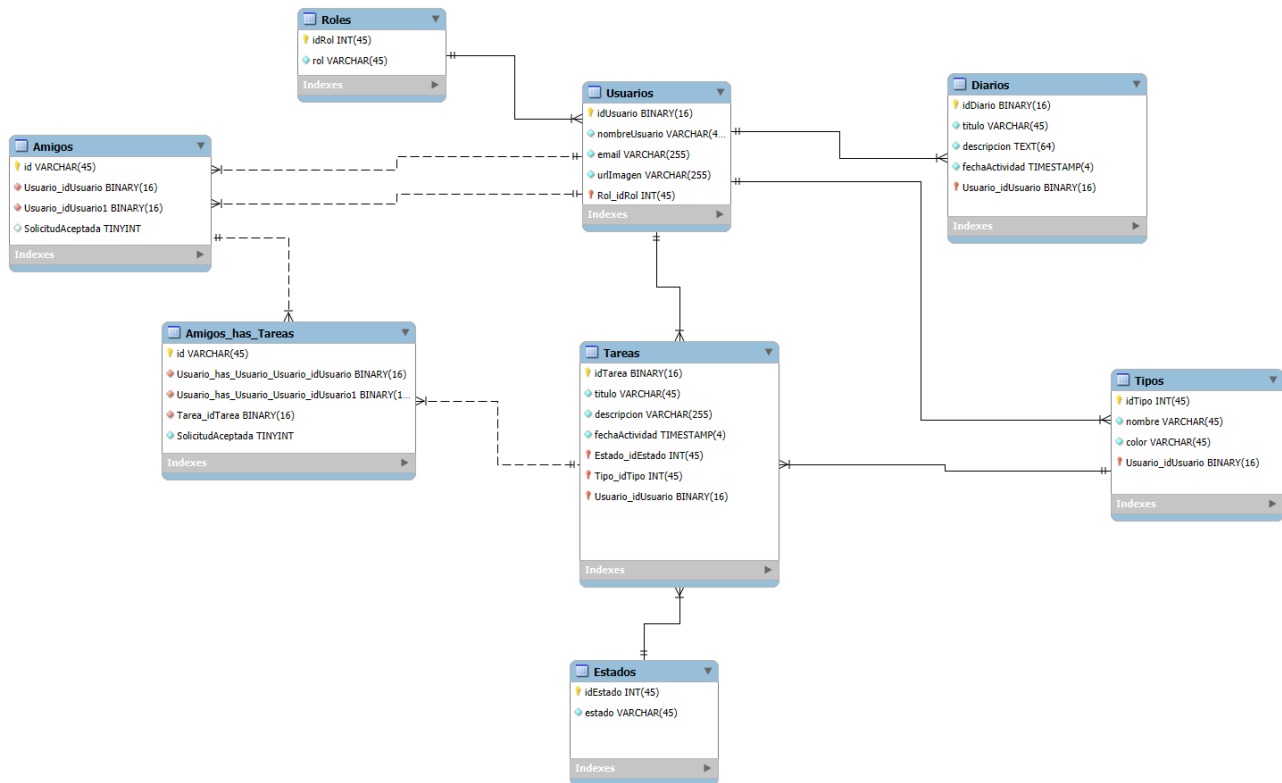
- Entidad tarea:
 - id: Clave primaria (Identificador único de la tarea).
 - título: Título de la tarea.
 - descripción: Descripción de la tarea.
 - fecha_actividad: Fecha para realizar la tarea.
 - categoría: Categoría de la tarea (por ejemplo, trabajo, personal, etc.).
 - id_estado: Clave foránea que hace referencia al id de la tabla estado. Esto establece que un estado puede tener varias tareas, pero cada tarea está asignada a un solo estado.
 - id_tipo: Clave foránea que hace referencia al id de la tabla tipo. Esto establece que un tipo puede tener varias tareas, pero cada tarea está asignada a un solo tipo.
 - id_usuario: Clave foránea que hace referencia al id de la tabla usuario. Esto establece que un usuario puede tener varias tareas, pero cada tarea está asignada a un solo usuario.
- Entidad tipo:
 - id: Clave primaria (Identificador único del tipo).
 - nombre: Nombre del tipo (por ejemplo, "Urgente", "Personal").
 - color: Color asociado al tipo.
 - id_usuario: Clave foránea que hace referencia al id de la tabla usuario. Esto establece que un tipo puede ser creado por un usuario, y cada usuario puede crear varios tipos.
- Entidad estado:
 - id: Clave primaria (Identificador único del estado).
 - estado: Nombre del tipo (por ejemplo, "Completo", "Incompleto").
- Entidad diario:
 - id: Clave primaria (Identificador único del diario).
 - descripción: Descripción del diario.

- fecha: Fecha en la que se creó el diario.
- id_usuario: Clave foránea que hace referencia al id de la tabla usuario. Esto establece que un usuario puede tener varios diarios, pero un diario solo puede pertenecer a un único usuario.
- **Relaciones:**
 - Usuario - Tarea: Un usuario puede tener varias tareas (relación de 1 a N). Cada tarea pertenece a un solo usuario.
 - Usuario - Diario: Un usuario puede tener varios diarios (relación de 1 a N). Cada diario pertenece a un solo usuario.
 - Usuario - Usuario (supervisor): Un usuario puede hacer amigos (otros usuarios) (relación autorreferencial). No todos los usuarios pueden tener amigos, por lo que esta relación es opcional.
 - Usuario - Tipo: Un usuario puede crear varios tipos (relación de 1 a N). Cada tipo pertenece a un solo usuario.
 - Rol - Usuario: Un rol puede asignar a varios usuarios (relación de 1 a N). Cada usuario pertenece a un solo rol.
 - Tipo - Tarea: Un tipo puede estar clasificada por varias tareas (relación de 1 a N). Cada tarea pertenece a un solo tipo.
 - Estado - Tarea: Un estado puede estar clasificada por varias tareas (relación de 1 a N). Cada tarea pertenece a un solo tipo.
- **Indices:**
 1. Claves primarias
 - Tabla rol: Índice en id.
 - Tabla usuario: Índice en id.
 - Tabla tarea: Índice en id.
 - Tabla tipo: Índice en id.
 - Tabla estado: Índice en id.
 - Tabla diario: Índice en id.
 2. Claves foráneas



- Tabla usuario:
- Índice en id_rol (relación con rol).
- Tabla tarea:
- Índice en id_usuario (relación con usuario).
- Índice en id_estado (relación con estado).
- Índice en id_tipo (relación con tipo).
- Tabla tipo:
- Índice en id_usuario (relación con usuario).
- Tabla diario:
- Índice en id_usuario (relación con usuario).

Modelo Relacional



Tablas

- **Roles**
 - id_rol → id_rol. Identificador único para cada rol.
 - nombre_rol → nombre del rol. (ejemplo: Administrador, Usuario)
- **Usuarios**
 - id_usuario → id_usuario. Identificador único del usuario
 - nombre → nombre. Nombre del usuario
 - email → email. Correo del usuario
 - contraseña → Contraseña cifrada del usuario para su autenticación.
 - id_rol → Relación con la tabla Roles, indicando el tipo de usuario.
- **Diarios**
 - id_diario → id_diario. Identificador único del diario
 - titulo → titulo del diario

- descripcion → Contenido o descripción del diario.
- fecha_creacion → Fecha donde se creó.
- id_usuario → Relación con Usuarios.
- **Tareas**
 - id_tarea → Identificador único de la tarea.
 - titulo → Nombre de la tarea.
 - descripcion → Detalles sobre la tarea.
 - fecha_vencimiento → Fecha límite para completar la tarea.
 - id_usuario → Relación con Usuarios, indicando quién la creó.
 - id_estado → Relación con Estados, indicando el estado actual de la tarea.
 - id_tipo → Relación con Tipos, indicando la clasificación de la tarea.
 - id_usuario → Relación con Usuarios.
- **Estados**
 - id_estado → Identificador único del estado.
 - nombre_estado → Nombre del estado (pendiente, en progreso, completada).
- **Tipos**
 - id_tipo → Identificador único del tipo de tarea.
 - nombre_tipo → Categoría de la tarea (trabajo, personal, urgente).
 - color_tipo → Color representativo del tipo de tarea en la interfaz.
 - id_usuario → Relación con Usuarios.
- **Amigos**
 - id_amigo → Identificador único de la relación de amistad.
 - id_usuario_1 → Usuario que envió la solicitud de amistad.
 - id_usuario_2 → Usuario que recibió la solicitud de amistad.
 - solicitudAceptada → Estado de la relación (booleano).
- **Amigos_has_Tareas**
 - id_amigo → Relación con Amigos, indicando qué amistad comparte tareas.



- `id_usuario_1` → Usuario que envió la solicitud de compartir tarea.
- `id_usuario_2` → Usuario que recibió la solicitud de compartir tarea.
- `id_tarea` → Relación con Tareas, indicando qué tarea está compartida.
- `solicitudAceptada` → Estado de la relación (booleano).

Relaciones

- **Usuarios ↔ Roles:** Cada usuario tiene un rol asignado.
- **Usuarios ↔ Diarios:** Cada usuario puede tener varios diarios.
- **Usuarios ↔ Amigos:** Gestiona las solicitudes de amistad entre los usuarios.
- **Usuarios ↔ Tareas:** Un usuario puede crear múltiples tareas.
- **Tareas ↔ Estados y Tipos:** Cada tarea tiene un estado y una clasificación.



5.5. SEGURIDAD Y AUTENTICACIÓN

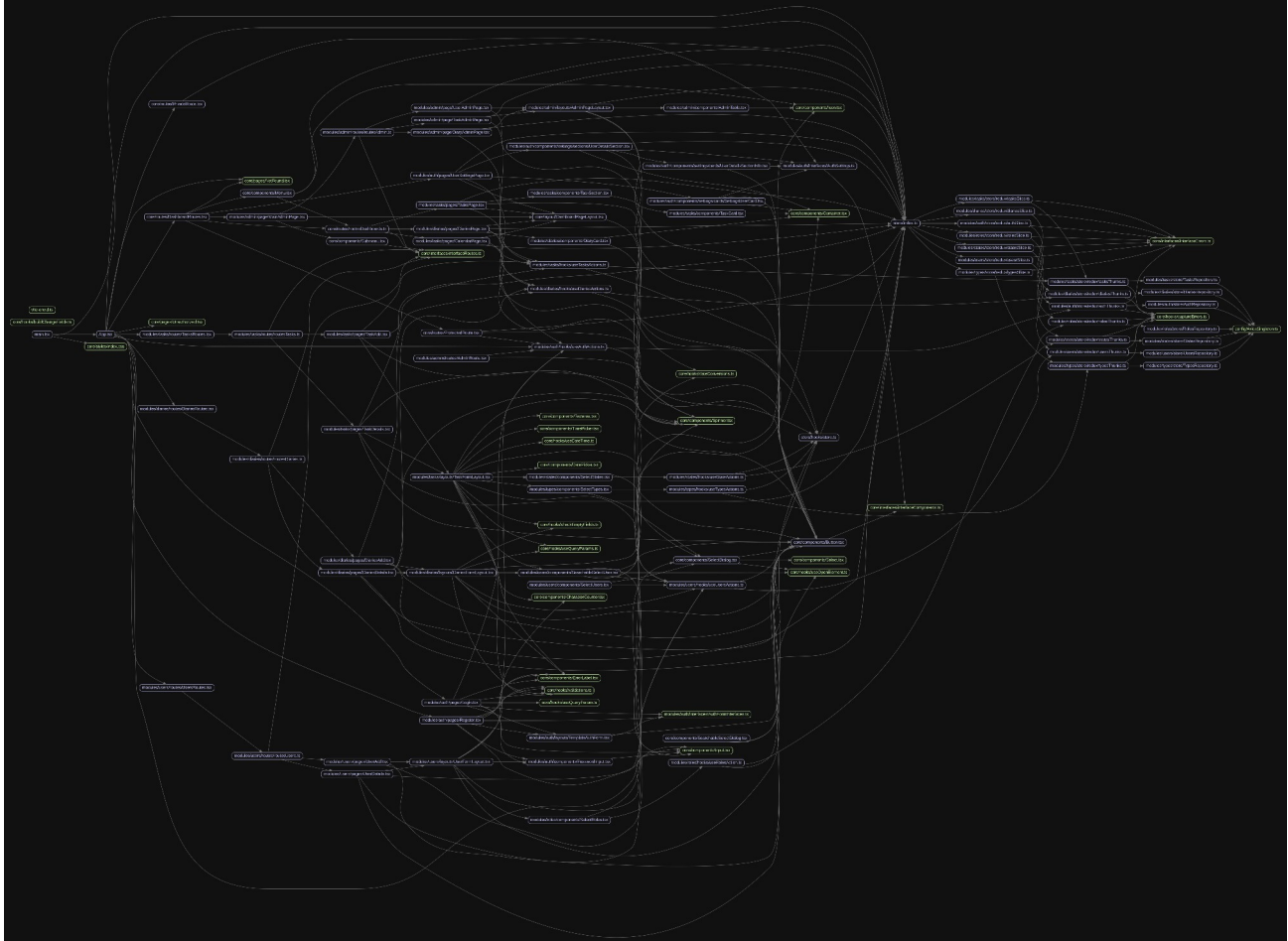
Para la seguridad de la aplicación se ha implementado una librería de **Node.js** para hashear contraseñas llamada **bcrypt**, el cual aplica un algoritmo unidireccional, lo que significa que no se puede revertir la contraseña original. Esta contraseña se guardará encriptada en la base de datos. Para la autenticación se ha utilizado **JWT (JSON Web Tokens)**, el cual permite crear y verificar tokens, para mantener sesiones. Este token se guarda en una **cookie** con el plugin **cookie-parser**, el cual solo el backend puede modificar. La combinación de ambas tecnologías es la siguiente:

- El usuario se registra → **hasheas** su contraseña con **bcrypt** y la guardas.
- El usuario inicia sesión → **comparas** la contraseña con **bcrypt.compare**.
- Generas un **JWT** y lo envías al cliente **dentro de una cookie HTTP-only y Secure**, para que **solo el backend pueda gestionarla** (ni JavaScript del cliente puede accederla).
- Luego, en cada petición protegida, el usuario envía el **token** en los headers (ej: Authorization: Bearer <token>), y tú lo verificas con **jwt.verify**.

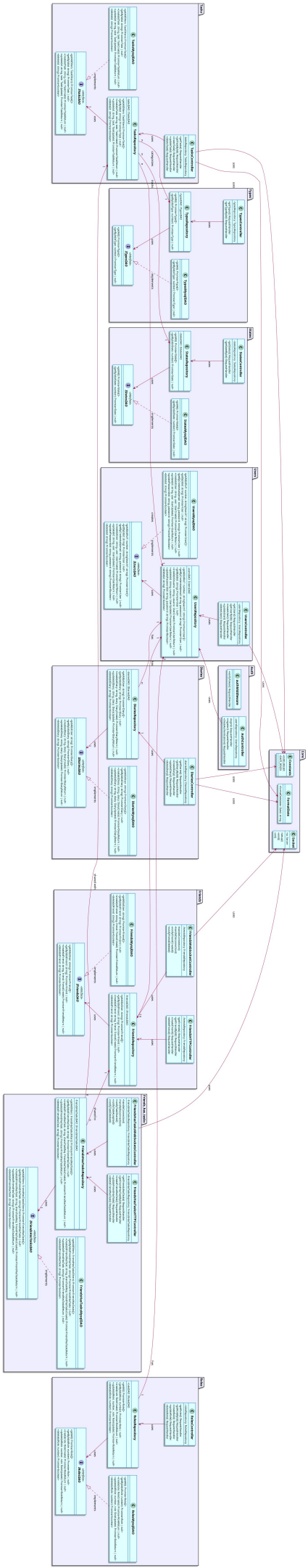
5.6. DISEÑO E IMPLEMENTACIÓN

DIAGRAMAS DE CLASES:

Frontend:



Backend:



EXPLICACIÓN DE LA ARQUITECTURA DE LA APLICACIÓN

La arquitectura de la aplicación sigue el modelo cliente/servidor, donde un cliente (frontend) se comunica con el servidor a través de una **API RESTful**. El servidor ha sido implementado mediante el uso de un patrón de diseño **MVC (Modelo-Vista-Controlador)**, el cual promueve una mejor separación de las responsabilidades y el mantenimiento del código fuente.

Cada entidad de la base de datos es gestionada a través de una **DAO (Data Access Object)**, el cual encapsula toda la lógica de acceso a datos, al igual que permite mantener separada la lógica de negocio del acceso directo a la base de datos. Cada carpeta del backend es estructurada siguiendo los principios de la **scream architecture**, donde cada funcionalidad tiene su propio módulo separado, por ejemplo usuarios, amigos, tareas...

IMPLEMENTACIÓN

El cliente realiza peticiones al servidor mediante el uso de **fetch**. En este caso para realizar esas peticiones de forma legible, se ha utilizado **Axios**, una librería de JavaScript/TypeScript, donde se envía datos o se solicitan recursos desde la interfaz de usuario. Estas peticiones son procesadas por el backend, el cual ejecuta la lógica correspondiente, consulta a la base de datos a través de toda la lógica de acceso a datos (DAO) y devuelve los datos necesarios.

HERRAMIENTAS:

- **Visual Studio Code (VSCode)** como entorno de desarrollo.
- **Postman** para probar y validar las rutas de la API.
- **GitHub** para el control de versiones del proyecto.
- **MySQL Workbench** como interfaz gráfica para gestionar la base de datos **MySQL**, conectada a **RDS de AWS**.

5.7. PRUEBAS Y CALIDAD DEL SOFTWARE

R1 – Inicio de sesión del usuario

- Se ha ingresado una dirección de correo y contraseña incorrectos y, después de ello, el sistema nos ha mostrado el mensaje de error. Prueba superada.

- Se ha ingresado una combinación que corresponde a unas credenciales correctas, las cuales se encuentran en la base de datos y, dado este hecho, se ha conseguido acceder al sistema de forma exitosa. Prueba superada.

R2 – Registro de usuario

- Se ha probado el formulario de registro, se han utilizado credenciales inválidas y se ha visualizado un mensaje informando del error. Prueba superada.
- Con credenciales válidas, el usuario se almacena correctamente en la base de datos. Prueba superada.
- Intentando guardar usuarios con datos inválidos se producen errores y, por lo tanto, no se registran. Prueba superada.

R3 – Gestión de Diarios.

- La petición Get devuelve la lista de diarios correctamente.
- La petición Post crea un diario en la base de datos con éxito.
- La petición Put actualiza correctamente un diario.
- La petición Delete elimina correctamente un diario.
- Al acceder, los datos del diario se cargan desde la base de datos y se muestran en pantalla (tanto para un nuevo diario como para uno que ya existía).
- Modificaciones realizadas se reflejan en la vista de la lista de diarios.
- La base de datos ha sido actualizada correctamente tras editar.

R4 – Gestión de Tareas

- La vista de la lista de tareas refleja correctamente los cambios.
- La vista de diarios refleja los cambios correctamente.
- La base de datos muestra correctamente la modificación.
- GET de tareas funciona correctamente.
- POST de tareas se crean correctamente.
- PUT modifica correctamente una tarea.

- DELETE elimina correctamente la tarea.
- Los datos de tareas se cargan correctamente en la pantalla.
- La información se muestra correctamente en pantalla.
- La base de datos refleja los cambios de edición.
- El calendario muestra correctamente la tarea modificada.
- La vista de tareas refleja correctamente los cambios de modificaciones.
- El calendario muestra correctamente las tareas editadas.
- La vista de tareas refleja correctamente los cambios de modificaciones.
- El calendario se actualiza correctamente con el dato modificado.
- Los cambios se reflejan correctamente en la vista de la lista de tareas.
- La base de datos refleja correctamente el cambio que esperamos.
- El calendario refleja correctamente la modificación.
- GET de tareas funciona correctamente.
- POST de tareas añade datos correctamente.
- PUT modifica correctamente los datos de tareas.
- DELETE elimina correctamente la tarea.
- El botón de acción en tareas funciona correctamente.
- El cambio se refleja en la base de datos tras la acción.
- El botón se comporta como se esperaba en la interfaz.
- La base de datos reflejará el cambio.

R5 – Gestión de Usuario

- GET de usuarios responde correctamente.
- POST crea nuevos usuarios sin errores.
- PUT actualiza correctamente los datos del usuario.
- DELETE elimina usuarios según lo esperado.
- La base de datos refleja el cambio de nombre.



- La interfaz para editar el nombre funciona correctamente.
- El cambio de contraseña se realiza correctamente.
- La interfaz de cambio de contraseña es funcional.
- El formulario de subida de imagen funciona correctamente.
- La imagen se guarda correctamente en el servidor.
- La URL de la imagen se ha establecido como foto de perfil.
- Los datos del usuario persisten tras recargar.
- La UI para cambiar email funciona correctamente.
- El email se actualiza correctamente en la base de datos.
- El botón 'Cerrar sesión' cierra correctamente la sesión.
- Se ha verificado que el usuario ha cerrado sesión.
- El botón 'Eliminar cuenta' responde como se espera.
- El usuario se elimina correctamente de la base de datos.

R6 – Aplicación web responsive

- **R6F1T1P1:** La aplicación se ha visualizado correctamente en diferentes tamaños de pantalla y dispositivos.

6. DESPLIEGUE DEL PRODUCTO

Se ha decidido utilizar servicios de Amazon Web Services (AWS) para desplegar el producto final. Esta decisión se ha tomado aprovechando aspectos como la fiabilidad, la escalabilidad y la disponibilidad de la solución. La arquitectura de despliegue queda repartida en:

Uso de EC2:

Se ha utilizado una **instancia EC2** ejecutando **Amazon Linux**, donde se aloja tanto el backend (API REST desarrollada en Node.js) como el frontend (aplicación SPA creada con Vite y React). Para garantizar una arquitectura estable y escalable, se ha utilizado el servidor web **Apache (httpd)** como **proxy inverso** y servidor de archivos estáticos.

El **backend** se ejecuta mediante un servidor **Express** que escucha por defecto en el puerto 3000. Para evitar que el proceso se detenga al cerrar la sesión SSH y garantizar su persistencia, se ha utilizado **PM2** como gestor de procesos:

El **frontend** ha sido construido con **Vite**, el cual te generará un directorio **/dist**. El contenido del directorio generado se ha copiado a la carpeta servida por Apache, el **/var/www/html/**.

La configuración de Apache cumple dos funciones:

1. Servir los archivos estáticos del frontend.
2. Redirigir las llamadas API a través de un proxy hacia el backend en localhost:3000.

Se ha creado y configurado un archivo **/etc/httpd/conf.d/frontend.conf**:

```
[ec2-user@ip-172-31-32-143 server]$ cat /etc/httpd/conf.d/frontend.conf
<VirtualHost *:80>
    DocumentRoot "/var/www/html"
    ServerName 13.60.87.209

    ProxyPreserveHost On
    ProxyPass /api http://localhost:3000
    ProxyPassReverse /api http://localhost:3000

    <Directory "/var/www/html">
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>

[ec2-user@ip-172-31-32-143 server]$
```

Al final la aplicación está alojada en esta url: <http://13.60.87.209/>.

Base de datos en RDS (MySQL):

El sistema de almacenamiento será Amazon RDS con motor MySQL. Esta base de datos relacional es totalmente gestionada por AWS lo que simplifica tareas como las copias de seguridad automáticas, la recuperación ante fallos, la actualización del software, la escalabilidad vertical... La instancia RDS quedará asociada de forma segura con el backend desplegado en EC2 con lo que se asegura baja latencia y que los datos quedan protegidos mediante configuraciones de red privada (VPC).

7. PRESUPUESTO, RIESGOS, VIABILIDAD

PRESUPUESTO

Horas de trabajo estimadas

1. Diseño
 - Diseño de Interfaz de Usuario(UI) y Experiencia de Usuario(UX): **28h**
2. Desarrollo
 - Desarrollo Frontend (Interfaz del Usuario): **20h**
 - Desarrollo Backend(Servidor y Base de Datos): **39h**
 - Integración y Sincronización de Datos: **10h**
3. Pruebas y Calidad
 - Pruebas de Funcionalidad: **15h**
 - Corrección de Errores y Optimización: **15h**
4. Despliegue : **16h**

Presupuesto total:

- $145h \times 14,5€/h = \mathbf{2102,5€}$ (El precio no incluye el servicio de AWS)

RIESGOS

Durante el desarrollo, han sido considerados los siguientes riesgos:

- Pérdida de información por fallo en el equipo o faltas de backups.
- Fallo en la integración de servicios externos (ejemplo APIs o servicios en la nube).
- Cambios de requisitos o alcance que podrían aparecer durante el desarrollo y que podrían influir en el cronograma.
- Saturación del servidor en el caso de un elevado crecimiento de usuarios.
- Problemas de Seguridad como inyecciones SQL, XSS y otras vulnerabilidades en caso que no se sigan buenas prácticas.

VIABILIDAD

La aplicación es viable ya que:

- Se utilizan tecnologías muy extendidas (Node.js, Express, bases de datos relacionales, etc.).
- Es fácilmente escalable gracias a la posibilidad de utilizar servicios como el que ofrece AWS.
- La interfaz está pensada para una buena usabilidad, basándose en pruebas y prototipos.
- Puede durar en el tiempo debido a su estructura modular y a su documentación suficiente.



8. DIFICULTADES ENCONTRADAS Y RESOLUCIÓN

Los desafíos que se han enfrentado durante el desarrollo del proyecto son:

1. **Test unitarios en el back-end:** la adaptación a los tests sumado a la inexperiencia en realizar testing hizo costosa la implementación de las pruebas
2. **Implementar los JWT:** al ser una tecnología que no se había visto, implementarlo en el proyecto fue una experiencia que tardó más de lo esperado.
3. **Implementar Compartir Tarea y Amigos:** se quiso implementar los Websockets en la aplicación para realizar solicitudes de amistad o de compartir tareas. En el backend se ha testeado y es funcional, pero falta implementar esa lógica en el frontend y testearlo ya que se priorizó otras funcionalidades para realizar la entrega del proyecto. Esto es una **mejora a futuro**.
4. **Subida de imágenes:** Al ser una implementación en la cual hay que configurar muchos servicios, se ha decidido no hacer para el día de la defensa para enfocarse en entregar una versión funcional del proyecto. Se tiene pensado también como **mejora a futuro**.

9. CONCLUSIONES

Llevar a cabo este proyecto ha permitido practicar los conocimientos adquiridos a lo largo del ciclo formativo de Desarrollo de Aplicaciones Web, atendiendo así todas las fases de la realización del software, desde la planificación inicial hasta el despliegue final en un entorno real de producción en la nube.

Con la culminación de este proyecto se ha conseguido crear una aplicación web útil, bien estructurada y funcional, que responde a una necesidad real: la gestión de tareas, diarios y relaciones personales a través de una herramienta digital de fácil comprensión. La elección de tecnologías como Node.JS, React y TypeScript y los servicios de AWS también se ha demostrado acertada y útil, para que se pueda garantizar un desarrollo de la aplicación seguro, escalable y eficiente.

A lo largo del proceso se han abordado varios retos técnicos, como la implementación de la autenticación JWT, la gestión de tareas compartidas o la estructuración modular del backend, todos ellos alcanzados con éxito. También se ha priorizado la usabilidad y accesibilidad de la aplicación, desarrollando una interfaz clara y adaptable a diferentes dispositivos mediante un diseño responsive.

A nivel personal, este proyecto ha supuesto un crecimiento significativo en cuanto a las habilidades técnicas, metodológicas y organizativas. Asimismo, ha evidenciado la relevancia de la planificación, la validación continua, el control de versiones y la documentación.

En conclusión, **TASK-CRAFT** no solo satisface los objetivos marcados, sino que también constituye una base robusta en la que seguir construyendo las mejoras futuras, tanto en lo técnico como en la experiencia de usuario final.



10. REFERENCIAS BIBLIOGRÁFICAS

Clear, J. (2020). Hábitos atómicos: cambios pequeños, resultados extraordinarios. adminAlex, & adminAlex. (2023, 13 marzo).

La gestión del tiempo y la productividad - Pro Optim Blog. Pro Optim Blog -.

<https://blog.prooptim.com/mejorar-productividad/la-gestion-del-tiempo-y-la-productividad/>

Midudev. (s. f.). *GitHub - midudev/curso-node-js: Curso de Node.js desde cero*. GitHub.

<https://github.com/midudev/curso-node-js>

Midudev. (s. f.). *aprendiendo-react/projects/10-crud-redux at master · midudev/aprendiendo-react*.

GitHub. <https://github.com/midudev/aprendiendoreact/tree/master/projects/10-crud-redux>

11. ANEXOS

MANUAL DE USUARIO

El presente manual contiene indicaciones para el uso adecuado de la aplicación. Las funcionalidades se organizan en secciones, e incluyen pasos a seguir para facilitar su uso.

Registro e inicio de sesión

- Para registrarse requiere:
 - Acceder a la pantalla principal.
 - Hacer clic en "Acceder".
 - Completar los siguientes campos:
 - Nombre de usuario
 - e-mail
 - Contraseña
 - Hacer clic en "crear una nueva cuenta".
- Para iniciar sesión:
 - Introducir el e-mail y la contraseña.
 - Pulsar "Acceder".
 - Si las credenciales son correctas se accede al panel principal.

Gestión de diario

- Para ver las entradas:
 - Acceder al menú lateral y seleccionar "Diario".
 - Aparece una lista con las entradas guardadas.
 - Para crear una nueva entrada:
- Pulsar "Añadir Diario".
 - Escribir el título y el contenido.
- Pulsar "guardar".



- Para editar o eliminar una entrada:
- Pulsar el botón de "editar" o "eliminar" al lado de la entrada deseada.

Gestión de tareas

- Para ver tus tareas:
 - Dirígete a la sección "Tareas" en el menú lateral
 - Revisa la lista con el estado de las tareas
 - Para crear una tarea nueva:
 - Pulsa "Nueva tarea"
 - Completa los siguientes campos:
 - Título
 - Descripción (opcional)
 - Fecha límite
 - Estado (pendiente, en curso, o completada)
 - Pulsa "Guardar".
 - Actualizar una tarea:
- Para actualizar una tarea, utiliza los botones de "Editar" para posteriormente modificarla.
 - Cambia el estado según el avance.
 - Pulsa el botón de eliminar para borrarla.
- Para filtrar tareas (no implementado):
 - Utiliza los filtros según el estado para ver tareas concretas.

Gestión de amigos (no implementado)

- Para buscar amigos y añadirlos:
 - Ve a "Amigos".
 - Introduce el nombre del usuario en la barra de búsqueda de amigos.
 - Pulsa "Enviar solicitud".



- Gestionar las solicitudes:
 - Puedes aceptar, rechazar solicitudes pendientes.
 - Ver amigos:
 - En la lista de amigos, selecciona "Ver perfil" o "Eliminar amigo".

Ajustes de cuenta

- Modificar el perfil:
 - Accede a "Ajustes" y modifica tu nombre, correo electrónico y contraseña.
 - Pulsa "Guardar cambios".
- Eliminar cuenta:
 - Ve a "Ajustes" y pulsa "Eliminar cuenta". Confirma la acción cuando aparezca.

Panel de administración (solamente para administradores de la plataforma)

- Acceder a la lista de usuarios registrados:
 - Acceder al "Panel de administración".
 - Visualizar la lista de usuarios, que incluye:
 - ID de usuario
 - Nombre del usuario
 - Rol que tiene asignado
 - Acciones posibles:
 - Cambiar el rol de un usuario
 - Eliminar cuentas de usuario

Navegación general

- El menú lateral permite navegar entre las distintas secciones: Diario, Tareas, Amigos, Ajustes.

- Los botones permiten identificar de manera clara el texto en cada uno de los, al menos, dos botones de los formularios.
- Los formularios validan los campos de los formularios antes de lanzar el formulario.
- Los formularios lanzan mensajes de error o confirmación para guiar a los usuarios.

LOGO DE LA APLICACIÓN:



TASK CRAFT

